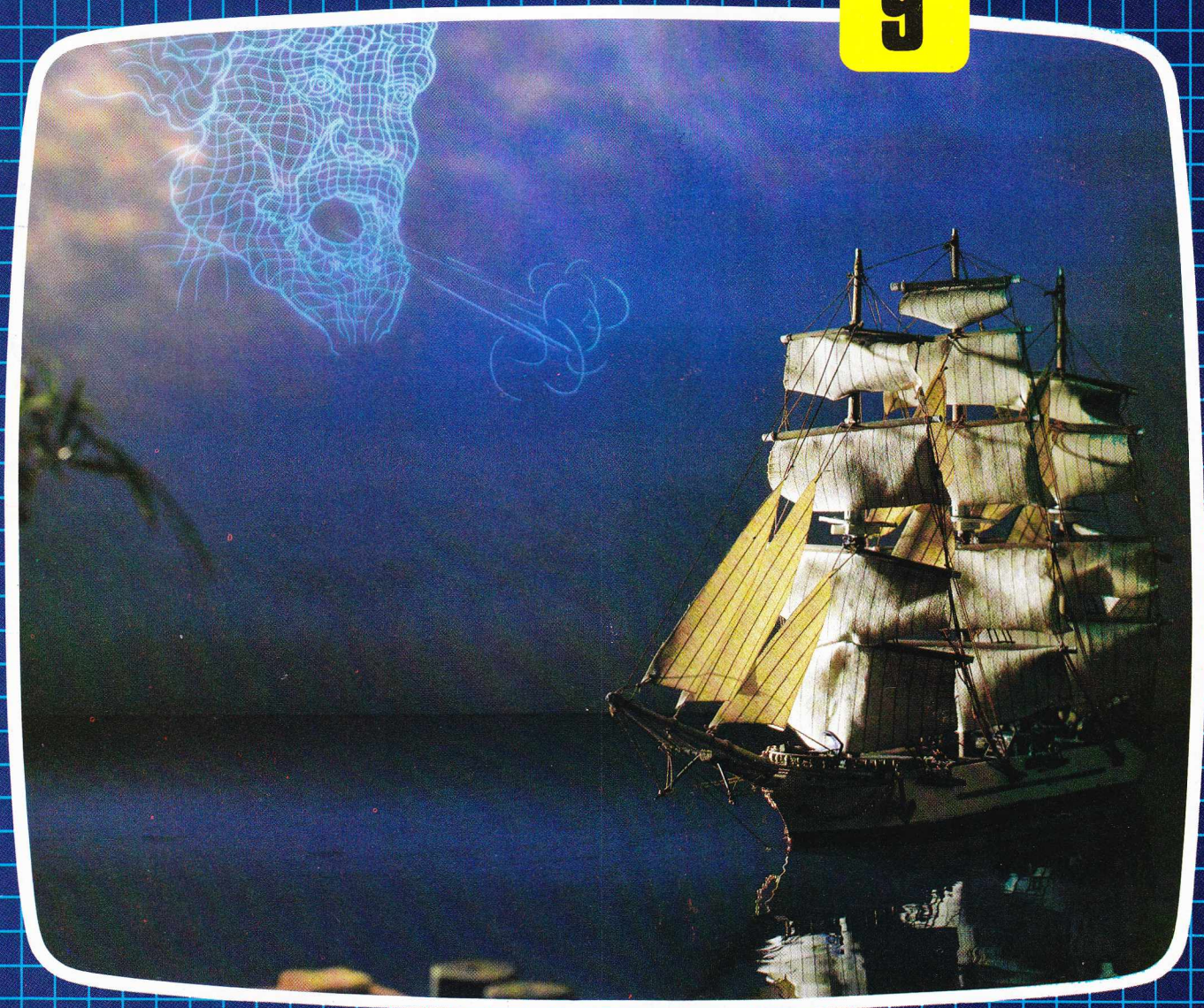
MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

9

Cr$ 8.200



TK 90X • O fator humano
Mouse e arte iconográfica

rioGráfica

MANAUS, ALTAMIRA, MACAPÁ, JI.PARANÁ, PORTO VELHO, RIO BRANCO, SANTARÉM, BOA VISTA, SINOP, ALTA FLORESTA: Cr$ 10.660

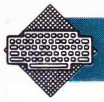
MICROCOMPUTADOR CURSO PRÁTICO

APLICAÇÕES

Paleta eletrônica 177

LINGUAGENS

Primeiras palavras 180



PERIFÉRICOS

Mouse e iconografia 184

RAIO X

TK 90X 186

SOFTWARE

Programas integrados 188

PROJETOS DE PROGRAMAS

A terceira dimensão 190

OFICINA

Teste de bancada 192

TÉCNICAS DE PROGRAMAÇÃO

O fator humano 194

PERFIL

Soluções integradas 196

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1) Qual foi a principal linguagem que originou o PASCAL?
2) Qual a expressão em inglês que qualifica um micro ou um programa que tem boa interação com o usuário leigo?
3) Qual o recurso disponível no Lotus 1-2-3 e no Symphony para armazenar seqüências de comandos freqüentemente utilizadas?
4) O que significa o símbolo Ω?
5) Qual o nome dos símbolos visuais que representam as diversas opções num menu de tela?

RESPOSTAS DO TESTE ANTERIOR

1) O programa de dump de tela imprime tudo o que estiver sendo mostrado no vídeo.
2) A grande desvantagem dos softwares integrados é que exigem grande quantidade de memória.
3) O principal recurso do BASIC para lidar com erros no programa é o comando ON ERROR GOTO.
4) O LOGO, como o BASIC, é uma linguagem interpretada, e não compilada.
5) Os procedimentos em LOGO que invocam a si mesmos ilustram o princípio da recorrência (recursion).

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 terá doze páginas e quatro capas, e incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as 3.ª e 4.ª trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 18 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem as 3.ª e 4.ª capas do fascículo 30); 3.ª e 4.ª capas dos fascículos de 1 a 29; e guardas.

rioGráfica **Diretoria Executiva:** Oscar Neves; Filipe Zander; Nilo Sérgio de Almeida; Danilo Esteves Costa; **GRUPO EDUCAÇÃO E CULTURA - Planejamento e Comercialização - Gerente de Marketing:** Jaime Rodrigues; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Ayiton Menezes; **Diretor de Vendas:** Marcos March; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Melio M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto; **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Mauricio Rittner; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Revisores:** Isis Augusta Loyolla; Cacilda Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Anibal Monteiro; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistente de Arte:** Miguel Luis Escámez Simón; **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; Claudio Marcondes; Nadya Morelo Reis; **Texto:** Paulo Brito; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Fernando Dorfman Knijnik; Lúcia Maria Leal Gonçalves; Maria Clara Cescato; Maria Ester Menezes Martinho; Newton Roberval Eichemberg; Patrícia de Paiva e Castro; Regina Amarante; **Pesquisa:** Stella Maria Quentel; **Arte:** Toshio Morobosi; **Assessoria Técnica:** Geraldo Coen; Gustavo José Ferreira Grubba; Lauro de Lauro; Pierluigi Pazzi; Imarés Microcomputadores. As fotos não creditadas pertencem à obra original. Seguem-se os créditos das demais: Dismac: Fernando Scavone; Jorge's Estúdio. APSIF Copenhagen, 1984; Orbis Publishing Ltd., 1984; Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** ERG Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Lda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

PALETA ELETRÔNICA

O tablete gráfico Koala existe em versões para o Apple e o IBM PC. Aqui examinaremos a versão para o Commodore 64, que utiliza plenamente os recursos gráficos de alta resolução desse microcomputador.

O Commodore 64 não foi suprido com periféricos para gráficos de alta qualidade, mas a companhia americana Koala Technology desenvolveu um tablete gráfico que permite aos usuários desse micro fácil acesso a sua capacidade de criar gráficos em alta resolução.

Medindo apenas 20,5 x 16 cm, o tablete gráfico Koala é leve e compacto. No centro, há um quadrado de fibra de carbono, de 11 x 11 cm, que cobre uma membrana sensível ao toque. Esta, pressionada com um dedo ou um lápis, guia o cursor pela tela.

Duas camadas de fios condutores — uma no eixo horizontal, outra no vertical — constituem a membrana. Quando ela é pressionada, o tablete gráfico detecta os fios que estão em contato e envia as coordenadas resultantes para o computador. Acima da membrana há dois botões, dos quais um deve ser pressionado quando o usuário deseja colorir um ponto da tela ou selecionar uma das várias opções de cores. Qualquer dos botões pode ser usado — provavelmente para facilitar o manuseio por canhotos e destros.

Colorindo

O tablete gráfico Koala conecta-se ao computador pela saída para joystick e o Koala Painter (o software necessário para o tablete funcionar) é carregado a partir de um disquete. Todas as opções disponíveis aparecem então na tela.

Na parte inferior fica a "paleta", contendo dezesseis cores "puras" e outras tantas "mistas". As misturas são obtidas colorindo-se pixels alternados com diferentes matizes, o que produz esse efeito.

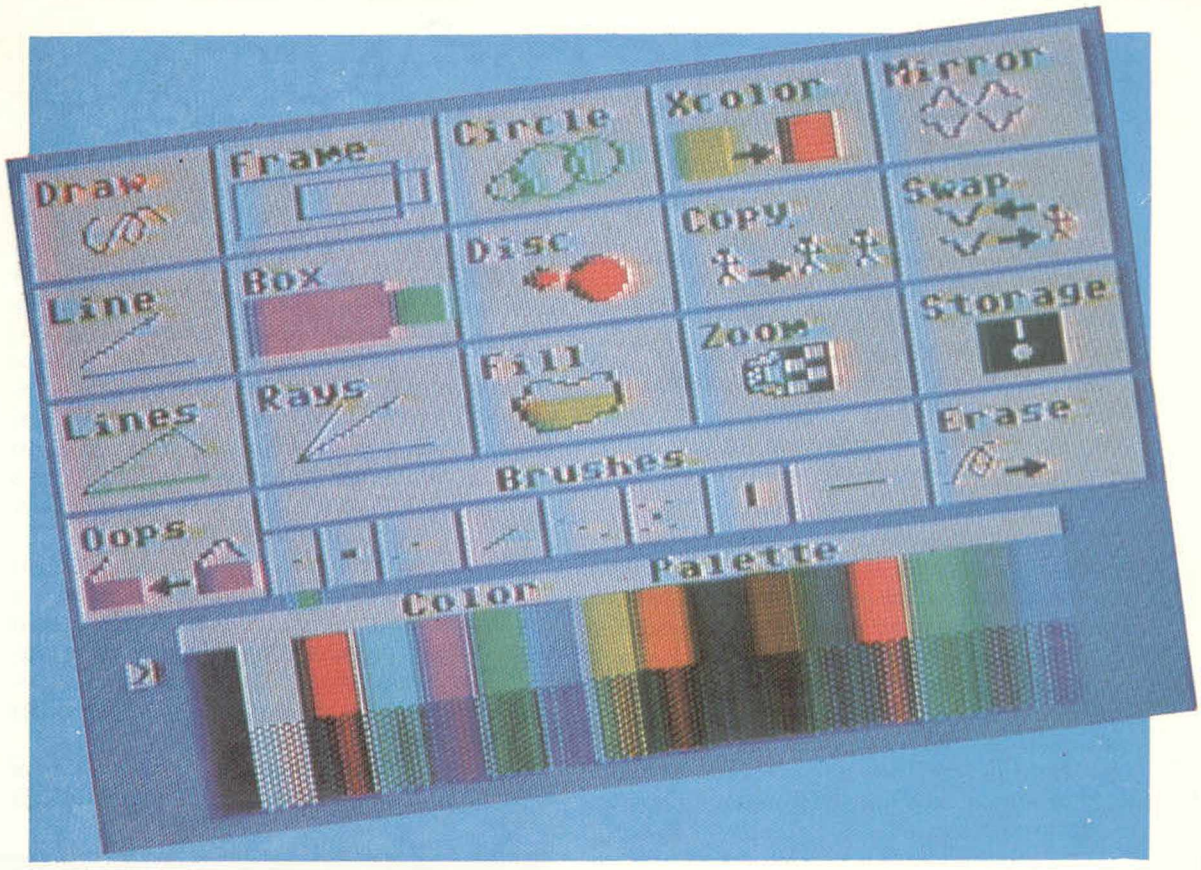
Acima da paleta há oito quadros contendo os "pincéis" (BRUSHES): formatos registráveis na tela, que variam de um único pixel até combinações de pixels e linhas. Ao redor dos pincéis há várias opções para o desenho de linhas ou formas na tela. O usuário as seleciona pressionando a membrana do tablete gráfico, direcionando um cursor em forma de seta. Quando a seta aponta para a opção desejada, a pressão num dos botões de seleção faz com que ela entre em ação. Uma luz intermitente brilha no retângulo indicativo da opção, para lembrar ao usuário qual o modo que está sendo usado.

O tablete gráfico Koala oferece recursos para produzir linhas isoladas, raios (linhas traçadas a partir de um único ponto), quadros e círculos. Blocos de cor podem ser acrescentados usando-se a opção BOX (quadrados coloridos) ou a opção DISC (círculos coloridos). Consegue-se manipular melhor o colorido usando o comando FILL para preencher uma área delimitada com a cor escolhida. As cores podem ser alteradas pelo uso do comando X-COLOUR.

Figuras idênticas são desenhadas simultaneamente usando MIRROR. Esse comando divide a tela em quatro seções, com o cursor restrito ao quadro superior esquerdo. Qualquer coisa traçada naquele quadro será automaticamente copiada nas áreas correspondentes dos outros três quartos.

Conseguem-se traçar desenhos detalhados a nível de pixel utilizando o comando ZOOM. O

Koala-pad
Telas de gráficos complexos e sofisticados são construídas em poucas horas quando se usa este software dirigido por menus de fácil compreensão.



Escolhas artísticas
O menu principal do tablete gráfico Koala consiste em quadros contendo o nome e o símbolo visual da opção. Apesar de os símbolos pretenderem auxiliar a compreensão, algumas das ilustrações são confusas. Para selecionar uma opção, move-se o cursor para um quadro e pressiona-se o botão de seleção. O nome do quadro selecionado ilumina-se intermitentemente para lembrar ao usuário que modo está sendo acionado.

usuário pode escolher qualquer trecho do desenho, que na parte inferior aparece como uma "janela" ampliada. Mostram-se quadrados do tamanho de um caractere de 8 x 8 pixels. Corrigidas, essas figuras são então colocadas em qualquer lugar da tela, por meio do comando COPY; isso permite definir na tela uma área cujo conteúdo é copiado em qualquer outra posição.

Selecionadas essas opções, o cursor é movido para fora da tela e o usuário pressiona o botão de seleção. A tela torna-se, então, o suporte para o quadro a ser pintado. Movendo o cursor e pressionando o botão de seleção, pode-se desenhar linhas e formas em qualquer parte da tela normal para gráficos do Commodore 64.

Prós e contras

A qualidade dos gráficos obtidos com esse dispositivo é excelente, alcançando o mesmo nível das telas de alta resolução produzidas por softwares comerciais.

No entanto, como em outros pacotes de gráficos, a qualidade do comando DRAW (para desenhos a mão livre) decepciona. A resolução da matriz da membrana não se equipara à da tela de alta resolução do Commodore 64 e, assim, o lápis (ou a ponta do dedo) do usuário raramente está sobre uma interseção da matriz. Na maioria das vezes atinge dois pontos ao mesmo tempo e o computador, ao interpretar isso, assinala um ponto que não se encontra na posição desejada.

Como resultado, não raro o que deveria ser uma linha reta acaba semelhando um rabisco desorganizado. Outra crítica é a de que, com exceção do comando ZOOM, não há opção para mudar de cor sem que seja necessário retornar ao menu principal. Mas essas são falhas secundárias, compensadas pela velocidade com que se executam os comandos LINE e FILL.

Outra limitação do software reside no método usado para eliminar erros, que emprega o comando OOPS, acessado a partir do menu principal. O uso do comando OOPS apaga todo o trabalho realizado pelo usuário desde a última vez que ele saiu do menu principal. Isso significa que meia hora de trabalho pode ser apagada devido a um único erro. Uma alternativa que se coloca é apagar um erro usando o comando ZOOM e corrigindo-o pixel por pixel, o que, no caso de um DISC ou BOX mal colocado, pode demorar algum tempo. Uma modificação bem-vinda seria restringir a extensão do comando OOPS à ultima pressão do botão de seleção e não mais à última saída do menu principal.

É possível usar o tablete gráfico Koala como um joystick, e assim se consegue acessá-lo a partir de seus próprios programas. A posição do cursor é obtida em BASIC pelo comando PEEK, usando as localizações 54297 e 54298 para as coordenadas x e y.

Após sua gravação em disco, as telas são facilmente transferidas para os programas do próprio usuário, o que permite o desenvolvimento de jogos com texto na parte inferior da tela e desenhos na superior.

Com o tablete gráfico Koala são gravadas e acessadas até dezesseis diferentes telas de 8 Kbytes num disquete. Como não é possível car-

regar uma tela diretamente a partir do disquete para a memória de tela, a Koala Technology fornece no guia do usuário um programa que possibilita transferir telas da área onde estão armazenadas para a memória de tela.

A resolução máxima de uma tela de Koala Painter gravada é de 255 x 255 pixels, se estiver incluída num programa em BASIC. Só se alcançam graus de resolução mais altos com o uso de linguagem de máquina. Essa limitação pode causar problemas, já que a tela de alta resolução do Commodore 64 tem o tamanho de 320 x 200 pixels.

O tablete gráfico Koala se restringe a um máximo de 255 pixels porque esse é o maior número que pode ser endereçado por 1 byte — o endereçamento de uma tela completa exigiria endereços de 16 bits. Uma das prováveis conseqüências disso é a perda de parte da imagem, pois, quando não está sendo exposta, a tela fica armazenada em um lugar determinado da RAM disponível para o usuário.

Portanto, a tela de 8 Kbytes, juntamente com as informações sobre as cores, deve ser transferida de sua localização na RAM do usuário para a memória de tela de alta resolução, sempre a partir da localização 55296.

O tablete gráfico Koala constitui investimento compensador para quem possui uma unidade de disco e gosta de arte ou de jogos de aventura.

Construindo um quadro

Esta série de etapas demonstra algumas das características do tablete gráfico Koala. Primeiramente, usando o comando ZOOM, o artista desenha cada letra da palavra "pleasure". Então, usando o comando COPY, ele as dispõe na parte inferior da tela, em seqüência. Depois, por meio do comando LINE, traça uma linha que atravessa o topo do letreiro e desenha o contorno da estrela para produzir efeito tridimensional. Blocos de cores são então acrescentados com o uso do comando FILL. Outras características puderam ser incluídas recorrendo aos comandos DISC e CIRCLE.

PRIMEIRAS PALAVRAS

Principais recursos do PASCAL

- Livre formatação do texto do programa.
- Regras flexíveis para nomear os objetos.
- Permissão de definir novas "palavras-chaves".
- Sintaxe simples e coerente.
- Definição dos dados.
- Estrutura de programação em módulos.
- Controle flexível dos dados e dos processos.
- Recorrência natural.
- Excelente diagnose de erro na compilação.
- Um compilador pequeno e muito eficiente.

O PASCAL é a mais influente linguagem de programação de alto nível. Embora algumas pessoas considerem sua sintaxe muito rígida, ele foi especialmente desenvolvido para o ensino da boa técnica de programação.

A linguagem PASCAL foi projetada pelo pesquisador suíço Niklaus Wirth, por volta de 1970. Seu nome é uma homenagem ao matemático e filósofo francês do século XVII, Blaise Pascal, o inventor da primeira calculadora com quatro funções. O PASCAL deriva diretamente do ALGOL 60 e constituiu uma resposta de Wirth ao ALGOL 68, uma linguagem por demais complexa e extensa. Ele pretendia fazer com que o PASCAL:

- permitisse a expressão exata dos conceitos e estruturas da programação;
- demonstrasse que uma linguagem concisa e de alto nível, dispondo de um conjunto flexível de tipos de dados, de instruções e de recursos para a estruturação de programas, pode servir como instrumento de uso geral para a solução de problemas;
- facilitasse a elaboração de métodos para organizar programas extensos e para manipular complexos projetos de software de modo seguro e garantido;
- tivesse amplos recursos para verificação de erros, especialmente durante a compilação, diminuindo assim ao máximo os erros de programação e fornecendo um veículo excelente para o ensino da programação; e
- pudesse ser usado eficientemente em micros.

Todos os objetivos do projeto foram plenamente alcançados — em geral, um pequeno compilador PASCAL ocupa apenas 24 Kbytes e é duas vezes mais eficiente que o FORTRAN (famoso por sua velocidade). Embora o PASCAL tenha um vocabulário restrito e seja fácil de aprender (possui apenas 35 "palavras-chaves" ou "reservadas", ao contrário do BASIC, com mais de uma centena, na maioria de suas versões), nem por isso é menos poderoso. O PASCAL é uma linguagem muito mais expressiva, tanto pelo modo de escrita dos algoritmos como pela facilidade para se descrever os dados de forma simples e coerente, seja qual for a complexidade destes.

Talvez a sua maior vantagem, no entanto, seja a de oferecer um método simples e coerente para a expressão de algoritmos extremamente poderosos. Afinal, o modo como consideramos os problemas de computação é o fator que mais facilita a sua resolução. A liberdade natural de expressão torna o PASCAL um excelente instrumento para a solução de problemas. Além disso, a linguagem apresenta vários outros recursos: por ser uma linguagem compilada, não apenas os programas são executados com muito mais rapidez, como também a memória não é desnecessariamente ocupada pelo programa-fonte e um interpretador de linguagem — mas apenas pelo código-objeto compilado.

Toda a filosofia da linguagem se resume em proteger o usuário de sua própria desatenção e evitar que rode um programa cheio de erros. Isso pode parecer estranho, especialmente para os programadores em linguagem BASIC. Muitas vezes, contudo, quanto mais rápido se elabora um programa em BASIC, mais tempo se leva para fazê-lo funcionar.

No caso de programas extensos é realmente muito mais fácil programar em PASCAL do que em BASIC. Embora muitas vezes o PASCAL seja criticado por suas regras estritas, são elas que o tornam tão eficaz na detecção e identificação de erros de programação. Afinal, essa disciplina imposta não passa de uma exigência necessária para a elaboração de qualquer programa com eficácia e segurança. Em conseqüência, economiza-se um tempo enorme que seria gasto na depuração de programas que "quase funcionam".

Para o programador habituado ao BASIC, a diferença mais notável dos programas em PASCAL, de qualquer extensão, é a grande quantidade de definições e declarações que parecem desprovidas de qualquer sentido. O primeiro 1,5 m de um programa extenso em PASCAL parece, de fato, não ter qualquer utilidade. Isso em parte resulta de que, embora você possa acrescentar à linguagem suas próprias palavras, todo o procedimento inicial é necessário para que o PASCAL entenda o que elas significam. Assim, enquanto no BASIC primeiro se define o programa principal (usando-se instruções do tipo GOSUB 5000) e, depois, as sub-rotinas, o PASCAL possibilita definir novos comandos como, por exemplo, LIMPARTELA ou PAUSA (tantos segundos) no início do programa e usá-los no procedimento principal. Por exemplo:

```
BEGIN
  LIMPARTELA;
  WRITE ('ALO!');
  PAUSA (3);
  ...ETC.
```

O PASCAL incentiva uma abordagem sistemática, mas muito prática, da programação de com-

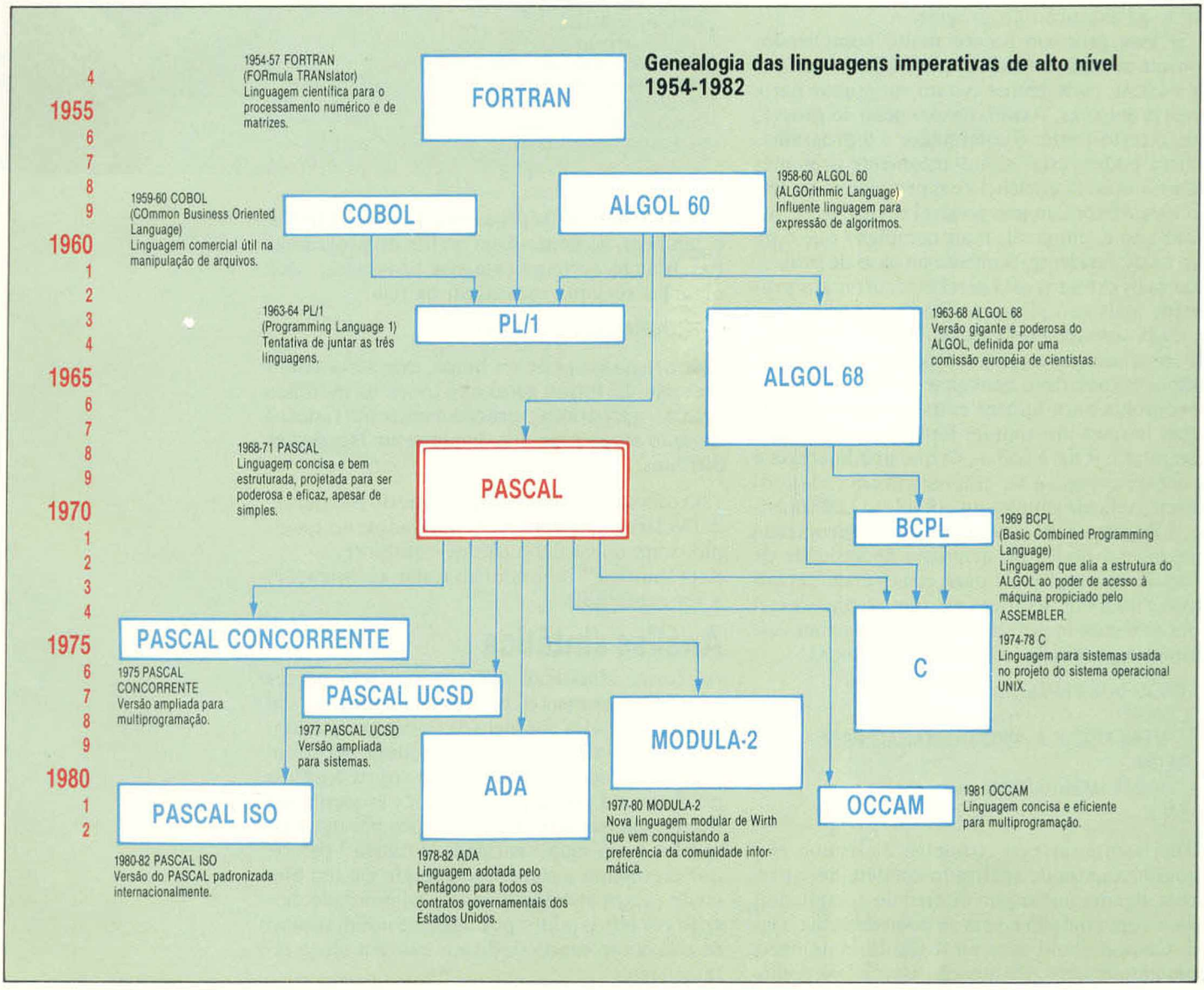
Blaise Pascal, 1623-1662
Matemático, cientista e filósofo, Blaise Pascal inventou a calculadora mecânica em 1642. O nome da linguagem PASCAL foi uma homenagem a sua contribuição para a ciência da computação.

putadores. Ele proporciona ao usuário uma visão conceitual de altíssimo nível a respeito de um sistema de computação, permitindo que dados e processos sejam definidos e expressos de modo natural e lógico. Isso garante um alto grau de portabilidade e segurança intrínsecos, e tanto a detecção de erros como seu diagnóstico são excelentes. De modo geral, o PASCAL teve mais influência sobre outras linguagens de computação — e no projeto de software — do que qualquer outra linguagem de programação.

O diagrama abaixo, que descreve a genealogia das linguagens de alto nível, mostra apenas as influências mais importantes das linguagens compiladas que empregam comandos imperativos; assim, não inclui as linguagens LISP, PROLOG, nem qualquer outra linguagem funcional. O tronco principal de influências tem início com o ALGOL 60 e será difícil encontrar uma única linguagem moderna digna de nota que não derive, direta ou indiretamente, do PASCAL. Conseqüentemente, um conhecimento sólido do PASCAL será muito útil para quem quiser entender as linguagens da década de 90 — como MODULA-2, OCCAM ou ADA.

Apesar de ter sido desenvolvido como um instrumento para o ensino de programação, o PASCAL foi adotado em muitos aplicativos e sistemas comerciais, e tem sido usado no projeto dos mais variados tipos de software, como pacotes financeiros e compiladores de linguagem. O sistema operacional "p-system", famoso por sua portabilidade e projetado no final da década de 70 pela UCSD (Universidade da Califórnia em San Diego), foi desenvolvido e escrito em PASCAL. O software para os micros Lisa e Macintosh, da Apple, inclusive seus sistemas operacionais, também foi elaborado principalmente em PASCAL ou em sua derivada, o CLASCAL. Em meados da década seguinte, a equipe da UCSD começou a projetar os compiladores para o MODULA-2 e outros sistemas, além de aplicativos para vários fabricantes de software — todos em PASCAL ou em linguagens derivadas.

O mais surpreendente é que tudo isso aconteceu sem qualquer investimento por parte de al-



guma empresa importante. O sucesso do PASCAL deveu-se apenas a seus próprios méritos e não por ser fonte de lucros para os grandes fabricantes.

Enfrentando o compilador

O primeiro problema que se encontra no uso de uma linguagem compilada está em aprender a lidar com o processo em vários estágios, mesmo que o programa seja pequeno. Em primeiro lugar, o programa-fonte deve ser introduzido mediante algum tipo de editor ou processador de texto. Gravado o texto-fonte em fita ou disco, o compilador da linguagem é carregado e recebe instruções para compilar o programa-fonte em código de máquina. Por fim, esse "arquivo-objeto" deve ser relocalizado, e suas partes — que foram compiladas separadamente — devem ser interligadas ("linkeditadas") às rotinas que serão necessárias para o processamento. Talvez o programa possa ser carregado e executado sem necessidade de trabalho complementar, mas se o compilador usar um pseudocódigo (código intermediário) será preciso valer-se de um interpretador na execução do programa.

Se esse processo parece muito complicado, console-se com o fato de que todas as versões de PASCAL para micros evitam em grande parte esses problemas. Assim, na execução do programa, o texto-fonte, o compilador e o programa-objeto podem estar simultaneamente presentes na memória. A eficiência e o pequeno tamanho do PASCAL tornam isso possível e o uso desse sistema não é, em geral, mais complexo que o de um BASIC residente. Somente no caso de programas mais extensos será preciso recorrer aos processos mais complexos acima descritos.

Cada sistema dispõe de seu próprio conjunto de comandos para controlar o editor e o compilador, e você deve consultar a documentação apropriada para utilizar esses recursos. Muitas vezes bastará um simples E para editar, C para compilar e R para rodar. O que nos interessa é a sintaxe correta a ser empregada em cada programa, seja ele simples ou complexo. Felizmente, a linguagem PASCAL é tão bem padronizada que haverá pouca ou nenhuma necessidade de adaptar os programas para cada versão, como ocorre no BASIC (embora existam algumas exceções de menor importância). Assim, examine esse primeiro programa completo em PASCAL:

```
PROGRAM PRIMEIRO (OUTPUT);
CONST
  MENSAGEM = 'PROGRAMACAO PASCAL';
BEGIN
  WRITE (MENSAGEM)
END.
```

Experimente carregar, compilar e executar esse programa antes de analisá-lo em detalhe. Se receber alguma mensagem de erro do compilador, leia-a com cuidado e veja se pode detectar o erro. Cada símbolo deve ser digitado exatamente como mostramos. Dê atenção especial ao ponto-e-vírgula no final da primeira e da terceira linhas e, também, ao ponto-final no fim do programa. Executando corretamente esse programa, você obterá a seguinte mensagem na tela:

```
PROGRAMACAO PASCAL
```

Esse programa pode ser banal, mas serve como exemplo da forma geral que todos os módulos PASCAL (programa, procedimento ou função) deverão apresentar. Ela divide-se em três partes distintas:

1. O cabeçalho; neste caso, o nome do programa.
2. Declarações e definições dos dados; no exemplo, uma única definição de constante.
3. O "núcleo", que contém todas as instruções a ser executadas.

Compilação popular
Uma implementação profissional completa do PASCAL pode muitas vezes custar mais que um micro, mas em meados da década de 80 foi lançada uma série de compiladores de preço relativamente acessível.

Análise sintática

As regras sintáticas do PASCAL, pelo menos quanto aos elementos básicos da linguagem, são melhor definidas por meio de "diagramas de sintaxe". Estes são como o mapa rodoviário de um sistema em sentido único. Uma rotina legítima no diagrama vai da parte superior esquerda até a parte inferior direita e todo bloco por que passarmos é: ou uma "entidade sintática" (isto é, que representa a si mesma) contida em um bloco de cantos arredondados, ou algum item descrito em outro ponto por meio de um diagrama de sintaxe separado (indicado por um bloco retangular).

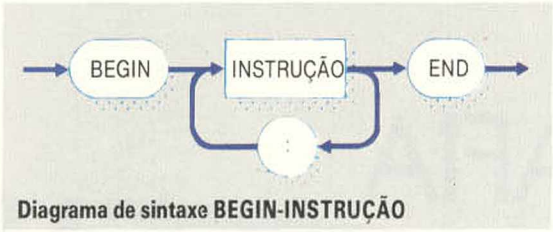


Diagrama de sintaxe BEGIN-INSTRUÇÃO

Com relação ao diagrama geral de um programa, note que as palavras BEGIN e END fazem parte do vocabulário reservado do PASCAL e não exigem diagramas complementares para definir seu significado. No PASCAL existem apenas 35 palavras reservadas com significado predeterminado. O programa acima utiliza somente quatro delas — PROGRAM, CONST (abreviação de constante), BEGIN e END. A palavra que segue PROGRAM é um "identificador", que indica o nome escolhido por você para o programa.

Se começarmos com uma letra e usarmos somente letras ou dígitos, teremos uma quantidade quase infinita de nomes aceitáveis. Entretanto, como era de se esperar, não é permitido usar uma palavra reservada. Por exemplo:

```
NOME
PASCAL
PROGRAMAUM
N
XYZ123
ENDERECO12
JOSESILVA
IDENTIFICADORBEMLONGO
```

são todas aceitáveis, enquanto:

```
PROG-1
DEZ%
AND
123QUATRO
SE PARECE
```

não são legítimas, ou porque contêm símbolos que não são alfanuméricos, ou porque começam por um número, ou ainda (no caso de AND) porque são uma das palavras reservadas. O último exemplo não é válido por causa do espaço entre as palavras, embora cada uma delas (SE e PARECE) seja aceitável. Em PASCAL, como na linguagem comum, não há diferença de significado entre letras minúsculas e maiúsculas, embora algumas versões não padronizadas exijam que as palavras reservadas estejam em maiúscula.

Além dos espaços e do final de linha, há um outro elemento da sintaxe do PASCAL que pode ser usado como um separador — o comentário. Este pode aparecer em qualquer ponto do texto, menos, é claro, no meio de palavras. Os comentários são indicados por chaves { }.

Sejamos um pouco mais ousados e examinemos algo mais parecido com um verdadeiro programa em PASCAL:

```
PROGRAM PROGRAMADOIS (INPUT, OUTPUT);
  {FORNECE O QUADRADO DE UM NUMERO}
CONST
  QUESTAO = 'ENTRE COM UM NUMERO:';
VAR
  NUMERO : INTEGER;
BEGIN
  WRITELN;
  WRITELN;
  WRITE (QUESTAO);
  READ (NUMERO);
  WRITELN (NUMERO, ' AO QUADRADO DO ',
    NUMERO * NUMERO)
END.
```

Observe que agora incluímos o identificador INPUT no título. O PASCAL exige os identificadores INPUT e OUTPUT, para a indicação dos arquivos externos com os quais o programa se comunicará. Geralmente, nos micros, eles serão o teclado e o vídeo, respectivamente. Com a inclusão do INPUT, podemos agora ler por meio do teclado, utilizando o procedimento padrão do PASCAL, READ. Como acontece com WRITE, qualquer "parâmetro" deve ser apresentado entre parênteses.

A memória utilizada para armazenar esses parâmetros é reservada por meio da declaração VAR — no exemplo, para um único número inteiro. Ao contrário do BASIC, que distingue somente entre números e strings (por meio do cifrão — $ — após o identificador), o PASCAL dispõe de um conjunto quase ilimitado de tipos de dados. Por isso, como o compilador precisa saber quanto de memória deve reservar para cada item dos dados, você sempre precisa declarar cada uma das variáveis em uma declaração VAR.

O cursor permanecerá posicionado imediatamente após o indicador do sistema, como se tivéssemos usado uma instrução PRINT do BASIC, com um ponto-e-vírgula no fim. Isso não é um erro, mas exatamente aquilo que solicitamos ao empregarmos o procedimento WRITE incorporado no PASCAL. Sempre que uma nova linha for necessária, usamos o procedimento alternativo WRITELN (LN é uma contração da palavra LINE). Uma instrução WRITELN simples, sem parâmetros, irá simplesmente criar uma nova linha.

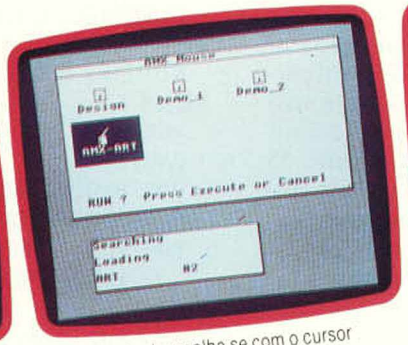
O PASCAL permite diferenciar entre dados que variam e itens que permanecem constantes durante a execução do programa. Assim, observe que a definição CONST usa um sinal de igualdade, enquanto a declaração VAR usa dois pontos.

As 35 palavras reservadas do PASCAL

AND	CASE	DO	END	FUNCTION	IN	NIL	OR	REPEAT	UNTIL
ARRAY	CONST	DOWNTO	FILE	GOTO	LABEL	NOT	PACKED	SET	VAR
BEGIN	DIV	ELSE	FOR	IF	MOD	OF	PROCEDURE	THEN	WHILE
							PROGRAM	TO	WITH
							RECORD	TYPE	

MOUSE E ICONOGRAFIA

O editor de ícones permite a criação de novos ícones ou a adaptação dos existentes.



Menu principal: escolhe-se com o cursor um dos quatro programas aplicativos.

Para criar uma figura, primeiramente se desenha o contorno.

Depois escolhem-se no menu os diversos tipos de linhas disponíveis.

Paralelamente à tecnologia do mouse, desenvolveu-se a fabricação do software que o acompanha. O AMX, por exemplo, inclui, além do mouse, o programa AMX Art, projetado para o BBC Micro.

O pacote AMX, da firma inglesa Advanced Memory Systems, criado para o BBC Micro, inclui o próprio mouse, acompanhado de um cabo para conectá-lo ao micro, um chip de ROM, dois manuais e software em cassete ou em disco.

Embora fornecido em cassete e em disco, o AMX é dirigido especialmente aos proprietários de micros com discos.

Fabricado no Japão, o AMX é um mouse comum, produzido em material plástico preto, com uma esfera de metal na base, para detecção de movimento. O mouse cabe na palma da mão, permitindo que três dedos se apóiem nos botões seletores de ação: Move, Execute e Cancel.

A decisão de incorporar uma esfera metálica — e não de borracha dura — mostrou-se contraproducente, pois esse material tende a escorregar em superfícies como madeira envernizada ou revestida em fórmica. Assim, é aconselhável colocar um pedaço de papel sobre a superfície lisa, para que a esfera role sem problemas.

Dentro do mouse, dois cilindros ladeiam a esfera. Na extremidade de cada cilindro há um disco com ranhuras que passa por uma célula fotoelétrica. O disco intercepta um feixe luminoso, criando impulsos ópticos captados pela célula e transmitidos como comandos direcionais ao computador. Desse modo, o computador sabe sempre a localização, o tipo de movimento e a velocidade do mouse.

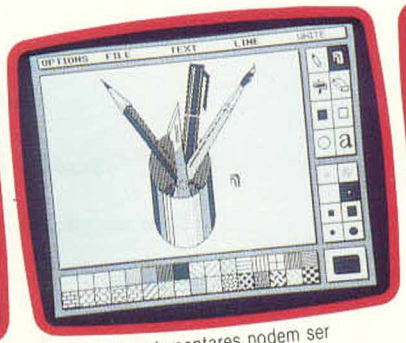
O software residente no chip da ROM situado na lateral faz a comunicação com o computador. Além das rotinas "primitivas" para uso dos pacotes de software, a ROM fornece várias rotinas que possibilitam a utilização do mouse em programas feitos pelo usuário. Essas rotinas incluem uma que fornece a posição do mouse, em coordenadas de tela; outra que lê os botões seletores; e uma terceira que anima os ícones (figuras) na tela. Usando essas rotinas — todas selecionadas pelo sistema operacional do BBC Micro —, o usuário pode escrever, com rapidez e facilidade, programas em BASIC que aproveitam ao máximo o controle do mouse.

Outras rotinas possibilitam o uso do mouse com pacotes comerciais de software. O movimento do mouse substitui as teclas do cursor e os três botões seletores podem ser configurados para agir como qualquer das três teclas, inclusive [Shift] e [CTRL]. Dessa forma, é possível usar o mouse para eliminar, mover e copiar partes de texto num processador.

No entanto, o valor do mouse só fica evidente quando se usa o software aplicativo desenvolvido para ele. O pacote AMX inclui dois progra-

Mouse AMX
O mouse AMX vem como parte de um pacote completo, que inclui o próprio mouse, um chip de ROM contendo o software de controle, vários softwares aplicativos em cassete ou disco, um manual do usuário e outro que ilustra aplicações artísticas.

Os contornos são preenchidos com uma série de motivos sombreados.



Texturas complementares podem ser acrescentadas.

Letras e palavras são adicionadas usando-se o modo texto e escolhendo o tipo desejado.

Retoques finais exigem o modo "a mão livre".

mas desse tipo, que demonstram a facilidade e a simplicidade do uso do mouse. O primeiro desses é um programa em BASIC para desenhar ícones incorporáveis em qualquer programa feito pelo usuário.

O programa de desenho Icon apresenta ao usuário um tabuleiro de dezesseis por dezesseis quadrados, nos quais ele poderá desenhar seu ícone preenchendo os quadrados desejados — tudo por meio do mouse.

Com esse programa, as únicas ocasiões para usar necessariamente o teclado ocorrem quando você for digitar o nome do arquivo em que gravou o grupo de ícones ou quando digitar o nome de arquivo de um conjunto já existente, para carregá-lo na memória. Devido ao alcance do controle do mouse, ele é fácil de usar e o desenho de ícones complexos e de efeito realista se faz com rapidez e simplicidade.

A melhor demonstração do potencial do mouse é dada pelo AMX Art, também incluído no pacote AMX. O AMX Art é um programa para desenho de gráficos com todos os recursos controlados pelo mouse. Equipara-se ao pacote Macpaint do Macintosh da Apple quanto à facilidade de utilização.

O uso do programa

Executa-se o desenho na região central da tela — o programa usa uma tela em duas cores, de 320 por 256 pixels. Do lado direito estão dois menus de ícones. O menu superior é empregado para escolher entre várias operações: desenhar linhas, apagar, esfumaçar e preencher áreas.

Cada uma dessas funções aparece na tela representada por uma imagem gráfica facilmente identificável: um lápis, uma borracha, um rolo de pintura. Determinada ação é escolhida movendo-se o cursor, por meio do mouse, sobre a imagem desejada e pressionando-se em seguida um dos botões seletores, o que dá ao cursor a forma do ícone escolhido.

A figura pode ser elaborada deslocando-se o cursor para a área a desenhar. Usando os botões seletores, você executa qualquer das funções do desenho. Como o movimento do cursor reproduz com precisão o movimento do mouse e o uso do programa é simples, o excelente manual que acompanha o AMX Art se torna quase desnecessário.

Outro menu que aparece na lateral da tela seleciona, pelo mesmo procedimento, as espessuras de linha e as formas. Na parte inferior da tela está um terceiro menu para a escolha de padrões sombreados que preenchem as formas. Eles variam do preto a complexos motivos entrelaçados. Como a tela só tem duas cores — devido a limitações de memória —, esses padrões são fornecidos para distinguir as áreas com diferentes tipos de sombreado.

Outras opções — como gravar ou carregar os desenhos criados, ativar os comandos do sistema ou listar um dump da memória — podem ser feitas por meio de menus que aparecem na parte superior da tela. Só as opções principais desses menus são apresentadas na tela; quando você posiciona o cursor sobre o item escolhido e pressiona um botão seletor, o restante do menu se desenrola verticalmente sobre a tela, podendo cobrir parte do desenho. Depois que você fez suas opções com o mouse, o menu desaparece, deixando a imagem como antes.

O usuário pode obter composições complexas, precisas e inovadoras já ao utilizar o programa pela primeira vez. Pouco tempo depois sente-se tão à vontade com esse pacote como se estivesse usando lápis e papel — e o AMX Art não quebra a ponta, não borra e pode ser reproduzido quantas vezes se deseje.

A AMX pretende ampliar o pacote mouse, transformando-o num sistema completo para o BBC Micro. Um gerenciador financeiro, semelhante no funcionamento ao que é usado nos computadores Macintosh e Lisa, estava sendo desenvolvido em meados da década de 80. A empresa também planejava um banco de dados controlado por mouse e outros aperfeiçoamentos para o programa AMX Art, incorporando cores e o efeito de zoom. O mouse e os menus pull-down serão com certeza incluídos em muitos programas aplicativos populares.

O mouse AMX e o programa AMX Art, assim como outros programas baseados no mouse e nos ícones, valorizam o BBC Micro para o usuário que estiver pensando em expandir os recursos de sua máquina.

RATOS, JANELAS E ÍCONES

Mouse: Dispositivo de operação manual conectado eletronicamente ao computador e dotado de esfera giratória na parte inferior. Movido sobre uma superfície plana, ocasiona um deslocamento correspondente do cursor na tela. O nome mouse (camundongo) deriva de sua aparência: uma caixinha, em geral preta, com um longo fio que se assemelha à cauda de um rato.

Menu pull-down: Menu que aparece na parte superior da tela, mostrando apenas as opções principais. Quando se seleciona um item, surgem as alternativas referentes a ele numa janela que se abre verticalmente sobre a tela. Terminadas as escolhas, o menu volta a ocupar apenas a linha superior da tela.

Ícones: Símbolos gráficos apresentados num programa para indicar as opções de um menu de modo visual, em vez de usar palavras ou combinações de teclas.

MEMÓRIA SELECIONADA

Dump: Listagem de todos os conteúdos de uma área da memória.

TK 90X

Irmão mais novo do TK 83 e TK 85, o TK 90X, da Microdigital, combina a vantagem de inúmeros softwares e periféricos disponíveis com comandos que facilitam a tarefa do usuário.

Portátil, capaz de funcionar ligado a aparelhos comuns de televisão, o TK 90X, da Microdigital, oferece imagens de alta resolução gráfica em oito cores básicas e geração de som pela tevê. Compatível em hardware e software com o ZX Spectrum +, da Sinclair, esse equipamento não apresenta nenhuma relação com seus antecessores TK 83 e TK 85. Considerado um avanço da linha Sinclair, o TK 90X incorpora uma interface para joystick — tipo Atari/Onyx Jr. —, com quatro posições de movimentação e uma de disparo, controladas também pelo teclado, e caracteres de acentuação para a língua portuguesa e cedilha. Tais recursos gráficos, no entanto, exigem o acionamento de três teclas.

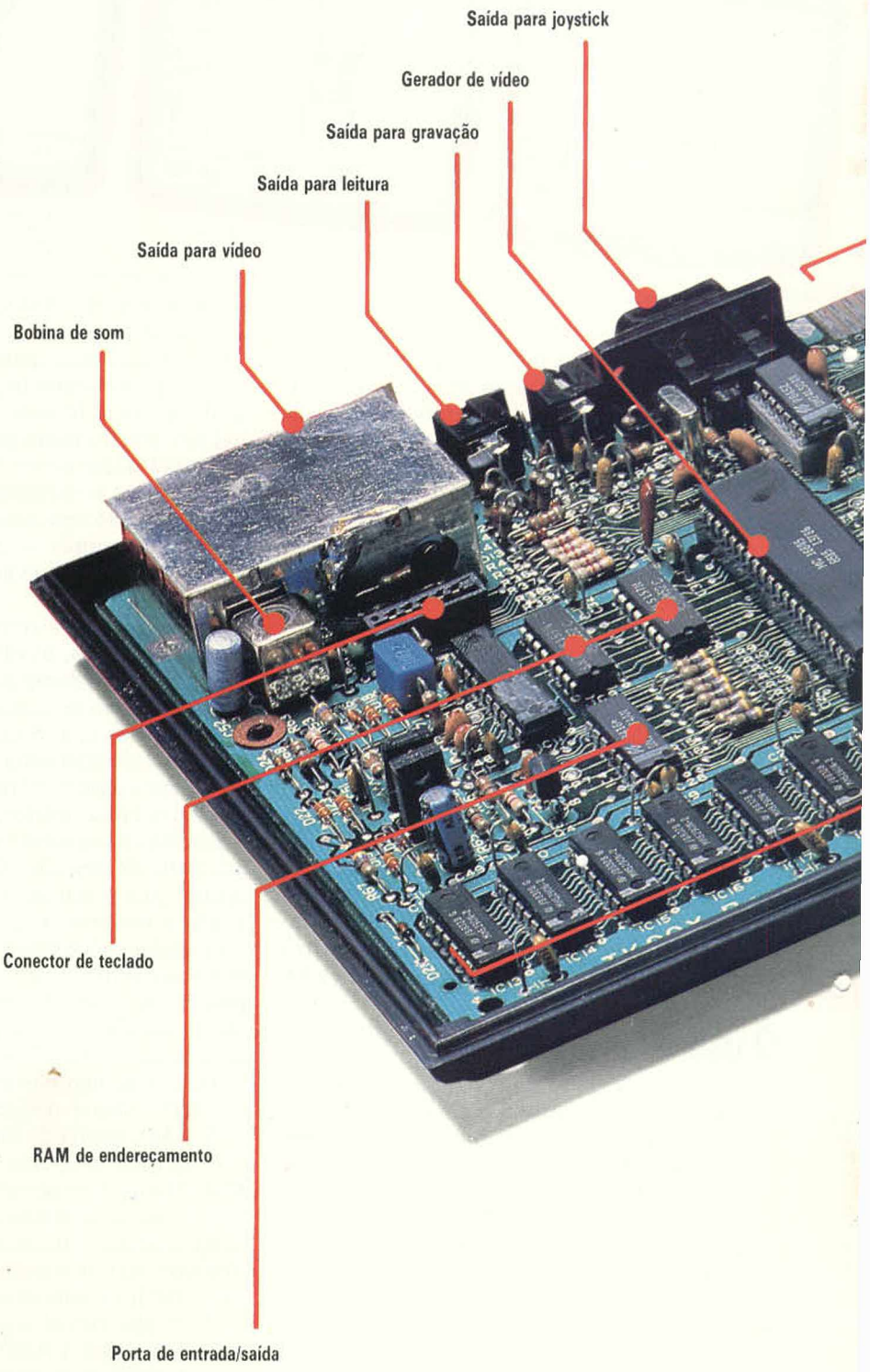
Esse modelo da Microdigital destaca-se também pela quantidade de periféricos disponíveis; possui saídas para Wafadrive, microdrive, unidade de disco, caneta óptica, gravador de EPROM, expansão de memória de 32 K, sintetizador de 10 oitavas, com 130 semitons, traçador digital e conversor analógico-digital.

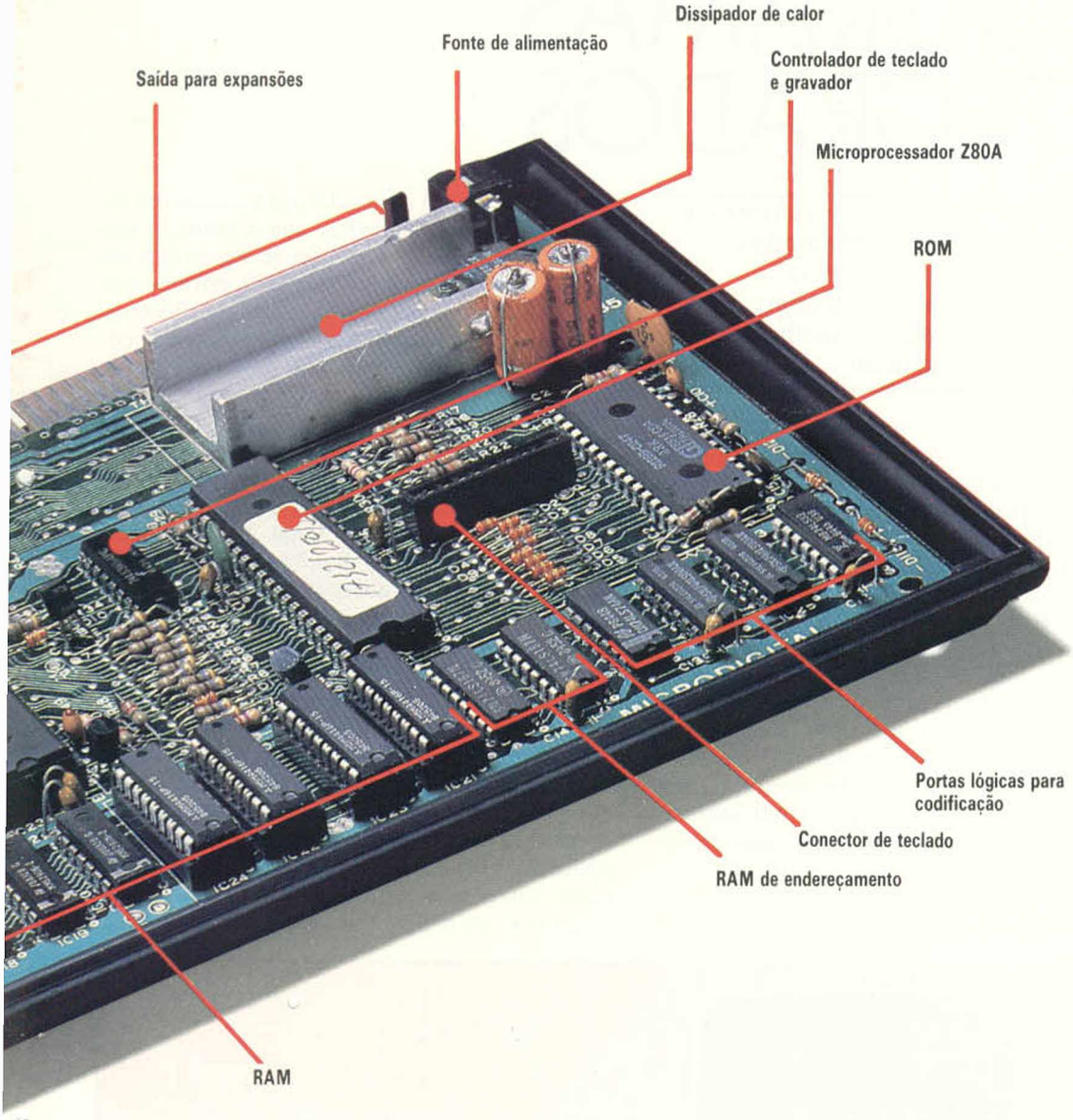
A grande quantidade de software disponível, no Brasil e no exterior, é outro ponto digno de nota. Os programas abrangem as áreas educacional, profissional e de lazer. O TK 90X roda também programas nas linguagens LOGO, PASCAL, FORTH e BASIC compilado.

O teclado tipo QWERTY compõe-se de quarenta teclas equivalentes a funções e comandos BASIC, compreendendo 37 funções matemáticas lógicas e científicas, além de 36 caracteres alfanuméricos, 26 caracteres minúsculos/maiúsculos, oito caracteres gráficos e mais oito em vídeo reverso.

O TK 90X apresenta diversos comandos que facilitam a tarefa do usuário. Para efeitos visuais, por exemplo, o comando INK permite escolher a cor do caractere mostrado no vídeo, enquanto o comando BORDER oferece a possibilidade de traçar margens coloridas, independentemente da cor no centro. Outro recurso é o comando FLASH (cintilador), usado para destacar informações na tela.

Utilizando fitas cassete comuns para armazenamento de dados, o TK 90X dispõe ainda de comandos que permitem ao usuário acompanhar um programa em andamento, bem como integrar dois ou mais programas BASIC na memória da máquina. Há comandos específicos para gravar e carregar programas, e também para ler dados e telas.





TK 90X

MICROPROCESSADOR

Z80A, de 8 bits.

CLOCK

3,58 kHz.

MEMÓRIA

16 K de ROM e 16 ou 48 K de RAM.

VÍDEO

Saída para tevê em cores, sistema PAL-M ou P&B. Modo texto: 24 linhas de 32 caracteres; modo gráfico: 256|x 192 pixels. Trabalha em modo invertido e apresenta recursos de flashing e brightness.

TECLADO

Tipo QWERTY, com quarenta teclas equivalentes a funções e comandos BASIC, 93 funções BASIC pré-programadas, 37 funções matemáticas lógicas e científicas, 36 caracteres alfanuméricos, 26 caracteres aifanuméricos minúsculos/maiúsculos, oito caracteres gráficos, oito caracteres gráficos invertidos, 21 caracteres gráficos criados pelo usuário.

LINGUAGENS

BASIC e ASSEMBLER.

PERIFÉRICOS

Programador de EPROM, expansão de RAM de 32 K, saída serial RS232C para comunicação de dados, saída paralela para impressora e saída para joystick.

DOCUMENTAÇÃO

Manual de operação e programa tutorial, em fita, que ensina a trabalhar com o teclado e a operar o micro.

Software diversificado
Compatível com o ZX Spectrum Plus, da Sinclair, o TK 90X executa grande número de programas — só na Inglaterra existem quatrocentos aplicativos educacionais para o aparelho, alguns dos quais já traduzidos pela Microdigital.

PROGRAMAS INTEGRADOS

Os programas Lotus 1-2-3, Symphony e Xchange são softwares integrados, projetados especialmente para microcomputadores profissionais. Mas suas técnicas se aplicam também a máquinas de uso pessoal.

O software integrado exige que o usuário tenha acesso instantâneo a todas as tarefas de que venha a precisar; e também que os procedimentos operacionais permaneçam inalterados, qualquer que seja a aplicação em uso; e, ainda, que as informações possam ser transferidas facilmente de uma aplicação a outra. Há várias maneiras de preencher esses requisitos.

O programa Lotus 1-2-3, da empresa americana Lotus, integra planilha financeira, banco de dados e geração de gráficos em cores. Operado por menus de fácil utilização, foi um campeão de vendas em todo o mundo, em 1984. Utiliza o formato da planilha financeira, na qual os números e as fórmulas são introduzidos em células definidas por linhas horizontais e colunas verticais, que podem ser modificadas.

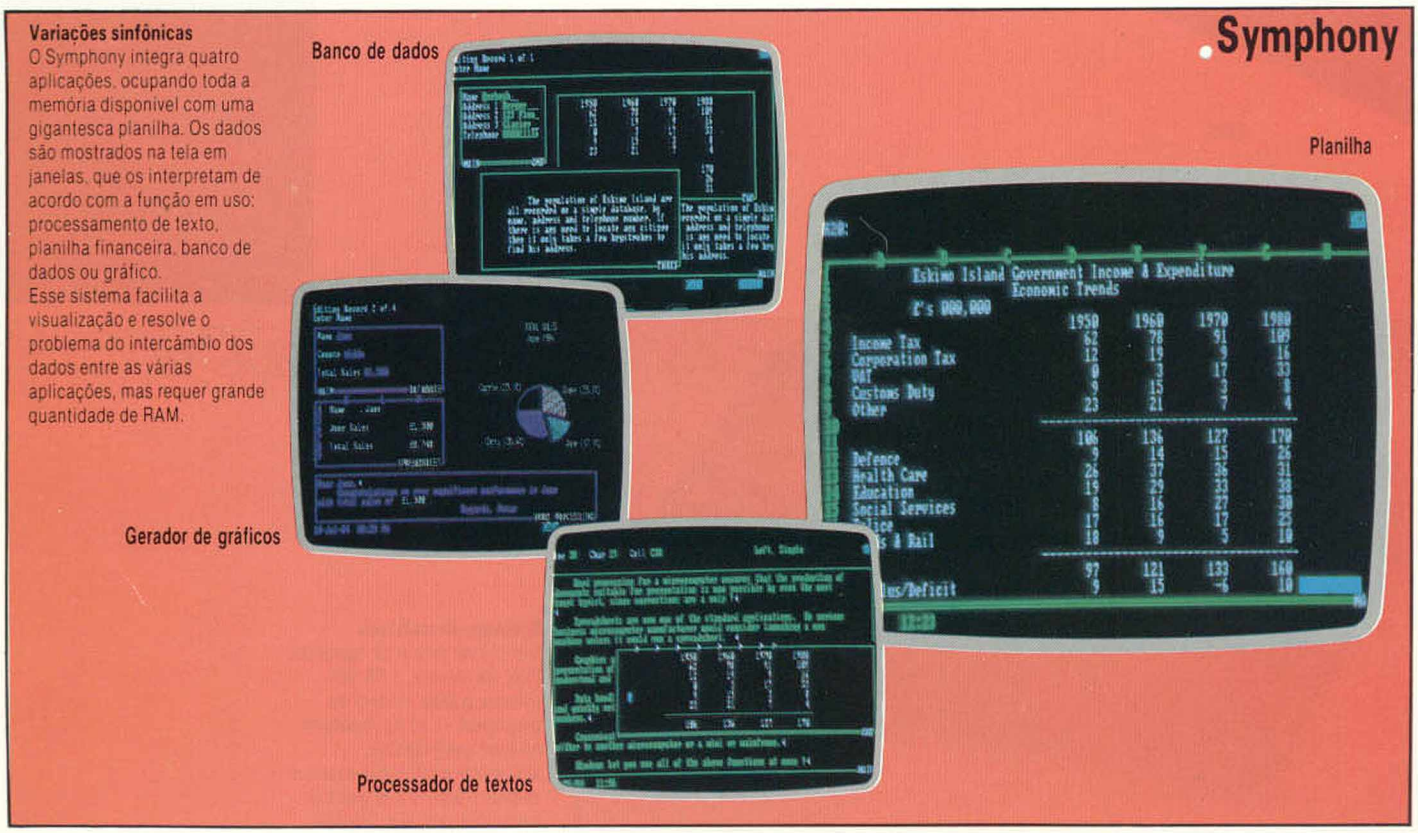
O programa oferece vários recursos extras — daí sua utilização para diversos fins. As células da planilha podem ser usadas para armazenar informações não financeiras, como nomes e números de telefone. Uma parte específica da planilha é utilizada como tabela de pormenores relevantes; por exemplo, uma lista de clientes e seus números de conta. Essa área será usada como um pequeno banco de dados, pois o Lotus 1-2-3 oferece funções de pesquisa e reorganização das informações.

Também é possível tomar uma série de células com dados numéricos e utilizar o programa para mostrar essa informação com diferentes tipos de gráficos em cores: linhas, barras, barras superpostas, ou círculo dividido em setores. Esse recurso elimina a necessidade de um programa gerador de gráficos separado.

O programa Lotus 1-2-3 tem ainda recursos para lidar com textos, o que permite seu emprego na redação de memorandos e cartas, mas a limitação de memória impede seu uso como um verdadeiro processador de texto.

Essa combinação de recursos faz do Lotus 1-2-3 o único programa necessário para grande número de usuários. Toda a informação que as aplicações requerem está contida numa planilha: assim, chega-se a resultados que seriam impossíveis com os programas tradicionais.

Variações sinfônicas
O Symphony integra quatro aplicações, ocupando toda a memória disponível com uma gigantesca planilha. Os dados são mostrados na tela em janelas, que os interpretam de acordo com a função em uso: processamento de texto, planilha financeira, banco de dados ou gráfico.
Esse sistema facilita a visualização e resolve o problema do intercâmbio dos dados entre as várias aplicações, mas requer grande quantidade de RAM.



Vamos supor, por exemplo, que um usuário do Lotus 1-2-3 administre várias bancas de jornais, em diferentes pontos da cidade. Ele precisa registrar as vendas semanais, mensais, trimestrais e anuais, referentes a cada uma das locações. A melhor forma de fazer isso é pôr a localização de cada banca e suas vendas numa planilha. As fórmulas são elaboradas de tal modo que o usuário só precisa mudar os números relativos às receitas semanais de cada banca. Os outros valores cujas fórmulas dependem daqueles números serão reajustados automaticamente.

Se o usuário quiser colocar as bancas em ordem de vendas — ficando a que vende mais no alto da lista —, encontrará recursos no Lotus 1-2-3. Inicialmente, introduzem-se os nomes das bancas em ordem alfabética, mas eles precisam ser reclassificados a cada semana, quando novos valores são recebidos. Com o Lotus 1-2-3, isso é rápido e fácil. O dono das bancas pode pedir ao programa um quadro semanal que mostre o desempenho de cada uma. Alguns toques no teclado permitem que essa informação seja acessada da planilha/banco de dados, mostrada na tela em forma de gráfico e impressa.

O programa da Lotus que se seguiu ao 1-2-3 — o Symphony — obedece ao mesmo princípio de basear as aplicações em formato de planilha, mas, além disso, permite a divisão da tela em diversas áreas (as "janelas"), focalizando diferentes partes da planilha. Cada janela ganha uma formatação apropriada, de acordo com a informação que mostra.

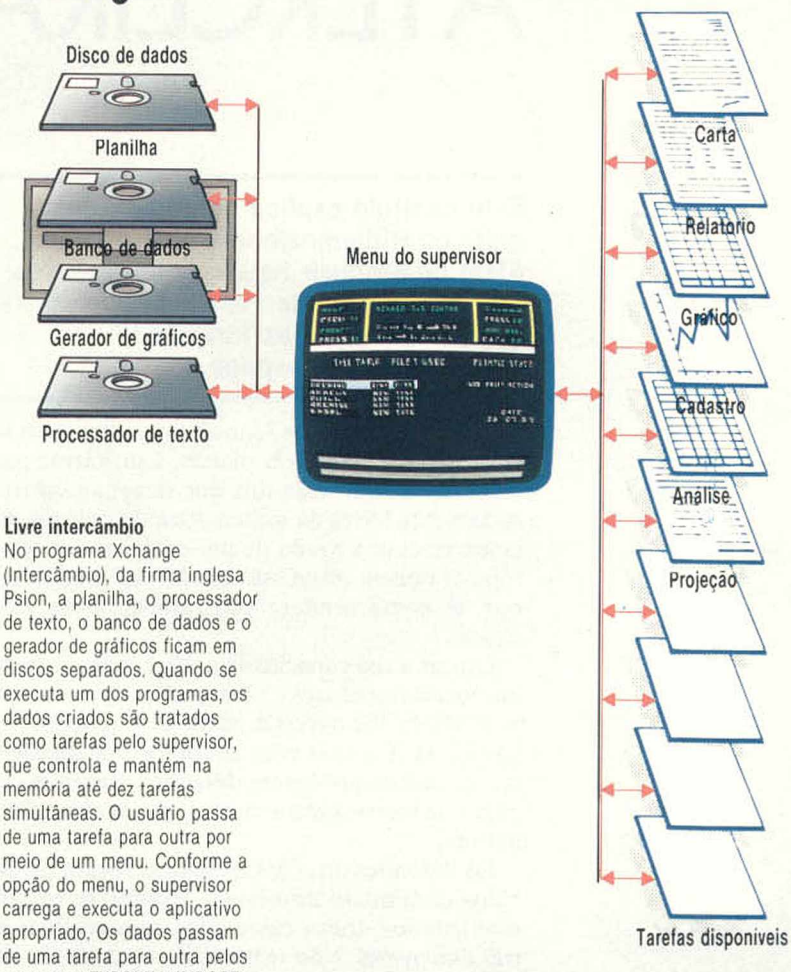
Quando a informação é um texto, a janela toma a forma de uma pequena tela de processamento de texto, com margens e tabulações bem demarcadas. Quando se precisa de um gráfico, outra janela mostra os eixos com suas escalas e seus títulos. Cada entrada de informação do banco de dados ganha sua própria tela, que se parece com um registro num fichário.

O Symphony tem quatro aplicações importantes, todas trabalhando ao mesmo tempo, e, assim, podem ser mostradas em separado na tela. Tal como o Lotus 1-2-3, memorizam determinadas seqüências de comandos, para que o usuário automatize as operações mais utilizadas. As seqüências são ativadas por pequenos programas chamados "macrocomandos".

O Symphony também inclui sua própria linguagem de programação de alto nível. Os programas são armazenados na planilha da mesma forma que quaisquer outros dados e acessam todas as operações disponíveis. Desse modo, se você precisar fazer um sistema de faturamento ou controle de estoque, pode programá-lo em linguagem Symphony. Os programas tornam-se parte de suas aplicações padrão.

Uma vez familiarizado com o Symphony, será mais simples programar em sua própria linguagem de comandos do que usar outra, pois o Symphony lida facilmente com tarefas como desenhar gráficos ou pesquisar e mesmo organizar dados.

Xchange



Livre intercâmbio
No programa Xchange (Intercâmbio), da firma inglesa Psion, a planilha, o processador de texto, o banco de dados e o gerador de gráficos ficam em discos separados. Quando se executa um dos programas, os dados criados são tratados como tarefas pelo supervisor, que controla e mantém na memória até dez tarefas simultâneas. O usuário passa de uma tarefa para outra por meio de um menu. Conforme a opção do menu, o supervisor carrega e executa o aplicativo apropriado. Os dados passam de uma tarefa para outra pelos comandos EXPORT e IMPORT.

O Symphony é apenas um dos muitos softwares integrados disponíveis no mercado e tem no Framework, da empresa americana Ashton Tate, um forte concorrente. Os dois são caros e exigem grande capacidade de memória. O primeiro opera com 320 Kbytes, mas precisa de 512 para o uso de todos os seus recursos. Já o Framework requer um mínimo de 256 Kbytes. Em função disso ambos só rodam em micros de 16 bits, como o IBM PC e compatíveis.

Nem o Symphony nem o Framework têm necessidade de que segmentos do programa fiquem armazenados em discos, caso da maioria dos programas aplicativos comerciais. Os chips de memória dos computadores continuam se tornando cada vez mais baratos; não é absurdo, portanto, que as equipes que desenvolvem software sustentem que os usuários terão grandes extensões de memória à disposição. Na prática, porém, isso ainda não acontece, e pode ser que tais aplicações com uso de memória extensa demorem algum tempo para se fazerem mais comuns. Embora um programa como o Symphony estabeleça novos padrões, ainda está sujeito às limitações do hardware. Em outras palavras, o Symphony só faz o que faz por ser um programa extenso e cuidadosamente elaborado.

A TERCEIRA DIMENSÃO

Este capítulo explica como construir gráficos tridimensionais com o micro. Além de produzir figuras interessantes, tal recurso permite visualizar o comportamento das funções matemáticas no espaço.

A simples menção de senos e tangentes, sem falar nos gráficos em três planos, é suficiente para balançar a segurança dos que se acham afortunadamente livres da matemática do colégio. No entanto, com a ajuda de um computador, esses tópicos podem tornar-se agradáveis, mesmo sem que se compreendam inteiramente seus princípios.

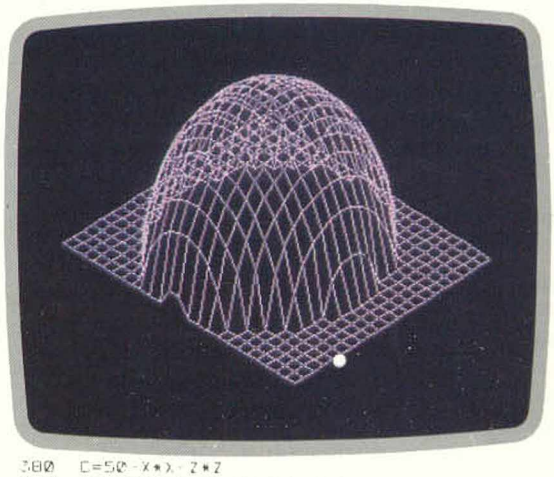
Graças a sua capacidade gráfica, a maioria dos microcomputadores é ideal para a apresentação de gráficos formados a partir de equações matemáticas. Escritas com símbolos matemáticos, tais equações produzem desenhos atraentes, capazes de interessar mesmo os que odeiam matemática.

Os desenhos aqui apresentados foram produzidos num micro doméstico, usando os programas listados, todos calculados com gráficos em três dimensões. Não se trata, na verdade, de gráficos tridimensionais, uma vez que os computadores só apresentam imagens em duas dimensões. No caso, "tridimensional" refere-se à ilusão de profundidade obtida com a superposição de diversos gráficos bidimensionais.

Os programas listados neste capítulo calculam os valores de uma equação com duas variáveis, X e Z. O resultado, Y, é calculado para cada valor de X e Z. Os valores de Y são usados para desenhar na tela os pontos situados no eixo vertical. Os pontos bem próximos ligam-se com pequenas linhas retas, que produzem, no conjunto, a impressão de uma linha curva. Um plano representa as coordenadas X e Y, mantendo-se Z constante, ao passo que o plano que o intersecciona perpendicularmente representa Y e Z, com X constante. A construção dos gráficos ajuda a entender funções matemáticas complicadas.

Esses gráficos também podem servir de estímulo a pessoas que não se interessam muito pela matemática. É divertido (e bem difícil) tentar montar uma equação que resulte numa determinada forma.

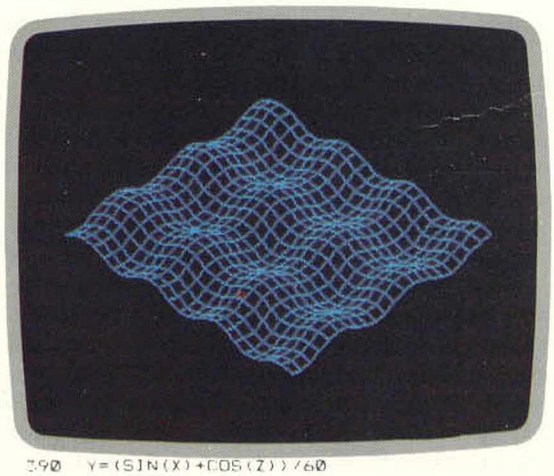
Para alterar o gráfico apresentado, é necessário mudar a função na linha 390 do programa em BASIC. Algumas funções são mais complicadas, exigindo, por isso, mais de uma linha de programa; se for esse o caso, pode-se usar todas as linhas com numeração entre 321 e 399.



```
380  C=50-X*X-Z*Z
390  Y=SQR(C*(SGN(C)+1))/30
```

Além de escolher uma função que resulte numa figura interessante, você deve cuidar para que os valores produzidos não sejam tão grandes que o gráfico ultrapasse os limites da tela. Para mantê-lo dentro dos limites, a função talvez tenha de ser dividida por um número grande.

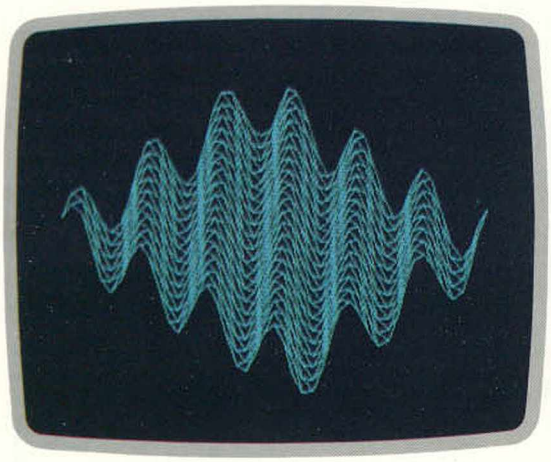
As versões deste programa podem ser usadas por microcomputadores de várias linhas. Para ajudar na conversão, o programa foi feito de modo que a primeira parte prepare uma diagramação padronizada da tela. Assim, uma equação que funciona em determinada máquina também deve funcionar em outras. A segunda parte do programa é usada para calcular e armazenar os valores que determinam os pontos do gráfico, mantendo-os numa matriz. Os cálculos dependem da função escolhida e podem demorar vários minutos; enquanto isso, o computador daria a impressão de que não está fazendo nada. Poupa-se tempo calculando a função antes. Se



```
390  Y=(SIN(X)+COS(Z))/60
```

Criação de formas
Insira ou modifique no programa básico as linhas mostradas embaixo de cada ilustração, a fim de produzir estes gráficos em três dimensões.

os cálculos fossem efetuados enquanto as linhas eram traçadas, o programa tomaria quase o dobro do tempo para traçar o gráfico.

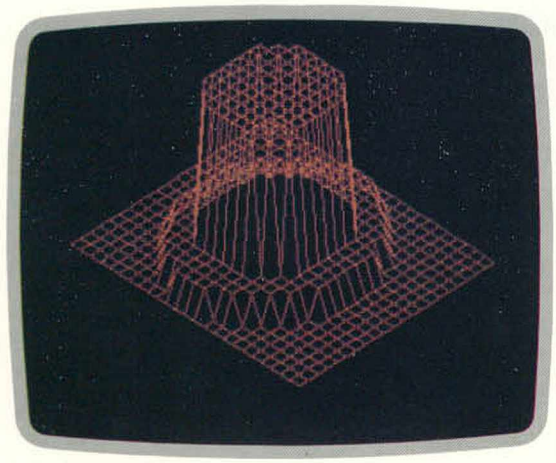


```
390  Y=SIN(X+Z)/24
```

Na listagem encontram-se diversas funções diferentes para você experimentar. As ilustrações mostram os resultados esperáveis. Você também pode tentar desenvolver seus próprios gráficos inserindo diferentes funções no programa. Tome cuidado ao fazer isso; é preciso ter certeza de que o gráfico caberá na tela e de que não será tentada nenhuma operação matemática ilegal. Os dois erros mais comuns são tentar dividir por zero (que dá infinito) e tentar extrair a raiz quadrada de um número negativo (que não é um número real).

Para evitar a divisão por zero, acrescente uma constante muito pequena (por exemplo, 0,00001) a qualquer variável que possa tornar-se zero. A única forma de proteger-se contra raízes quadradas de números negativos é usar a função ABS (valor absoluto) para tornar todos os números positivos antes de extrair a raiz quadrada.

Alguns gráficos interessantes resultam de funções matemáticas comuns, tais como SIN (seno), COS (co-seno) e LOG (logaritmo). Outros podem ser conseguidos usando funções só encontradas em computadores — experimente INT (inteiro), SGN (módulo) e ABS.



```
380  C=X*X+Z*Z+0.00001
390  Y=SGN (INT(13/C))/3+ SGN(INT(35/C))/15
```

Este programa aceita aperfeiçoamentos diversos. Você pode tentar adaptá-lo de maneira que qualquer função seja colocada em escala automaticamente, a fim de enquadrar-se nos limites da tela. Ou, também, marcar os pontos numa terceira direção, dando curvas para X e Z, enquanto Y é mantido constante (o que é mais complicado). Mas, mesmo usando o programa como está escrito, você verá que é muito divertido trabalhar com as mais simples equações que se pode imaginar. Os resultados certamente irão surpreendê-lo.

Figuras e planos

```
100  REM ********************************
110  REM **                              **
120  REM **   GRAFICO TRIDIMENSIONAL     **
130  REM **                              **
140  REM **       PARA O APPLE II        **
150  REM **                              **
160  REM ********************************
165  REM
170  LARGURA=280:ALTURA=190:VA=-1
180  XFAL=5:ZFAL=3
190  TAMANHO=INT(LARGURA/XFAL/2)
200  PROF=INT(ALTURA/ZFAL/3)
210  HOME:HTAB10:VTAB3:PRINT"CALCULANDO"
220  REM
240  REM ********************************
245  REM **       CALCULA GRAFICO        **
250  REM **    OBSERVAR TEXTO PARA       **
255  REM **         VARIACOES            **
260  REM ********************************
265  REM
270  INICIO=5
280  DIM G(TAMANHO,PROF)
290  FOR A= -PROF/2 TO PROF/2
300  FOR B= -TAMANHO/2 TO TAMANHO/2
310  VTAB23:HTAB03:PRINT"A>";A:
     VTAB23:HTAB12:PRINT"B>";B
320  X=A*18/TAMANHO:Z=B*18/PROF
390  Y=(SIN(X)+COS(Z))/60
400  G(B+TAMANHO/2,A+PROF/2)=Y*VA*ALTURA
410  NEXT B:NEXT A
420  REM
440  REM ********************************
450  REM **      DESENHA PLANO X-Y       **
460  REM ********************************
465  REM
470  HGR2:HCOLOR=7
480  FOR Z=1 TO PROF
490  XBASE=XFAL*Z
500  ZBASE=ALTURA/2+Z*ZFAL+INICIO*VA
510  XVELHO=XBASE+XFAL
520  ZVELHO=191-ZBASE+ZFAL+G(1,Z)
530  FOR X=1 TO TAMANHO
540  XNOVO=XBASE+X*XFAL
550  ZNOVO=191-ZBASE+X*ZFAL+G(X,Z)
560  HPLOT XVELHO,ZVELHO TO XNOVO,ZNOVO
570  XVELHO=XNOVO:ZVELHO=ZNOVO
580  NEXT X:NEXT Z
590  REM
600  REM ********************************
610  REM **      DESENHA PLANO Z-Y       **
620  REM ********************************
630  REM
640  FOR X=1 TO TAMANHO
650  XBASE=XFAL*X+PROF*XFAL
660  ZBASE=ALTURA/2-X*ZFAL+PROF*ZFAL+
     INICIO*VA
670  ZVELHO=191-ZBASE+ZFAL+G(X,PROF-1)
680  XVELHO=XBASE-XFAL
690  FORZ=0 TO PROF-1
700  XNOVO=XBASE-Z*XFAL
710  ZNOVO=191-ZBASE+Z*ZFAL+G(X,PROF-Z)
720  HPLOTXVELHO,ZVELHO TO XNOVO,ZNOVO
730  XVELHO=XNOVO:ZVELHO=ZNOVO
740  NEXT Z:NEXT X
750  GET FIM$
760  TEXT:HOME:END
```

TESTE DE BANCADA

Ligar um aparelho para ver se funciona implica o risco de contato acidental de dois circuitos condutores. Evite danificar delicados componentes montando um aparelho de teste e um multímetro.

A primeira etapa na verificação de qualquer montagem de componentes num circuito testa a eficiência das soldas nas ligações. Às vezes, uma solda feita a temperatura muito baixa parece aceitável, mas não está fazendo nenhuma ligação interna. Um ligeiro puxão talvez revele isso. Se você a executou corretamente, a ligação não se partirá em suas mãos; mas, se isso acontecer, será melhor que ocorra nessa etapa.

Pode-se encontrar o tipo mais simples de aparelho de teste de contato em lojas de ferramentas e de acessórios para automóveis. Vendido com o nome de "teste de ignição", ou outro similar, é utilizado para determinar se o contato no distribuidor está aberto ou fechado.

Para manutenção e pequenos reparos em micros, a garra jacaré — que em geral se encontra num dos terminais do aparelho — não é muito útil. Elimina-se esse inconveniente soldando um pedaço pontiagudo de ferro na garra, junto com fita isolante, e usando-o como teste.

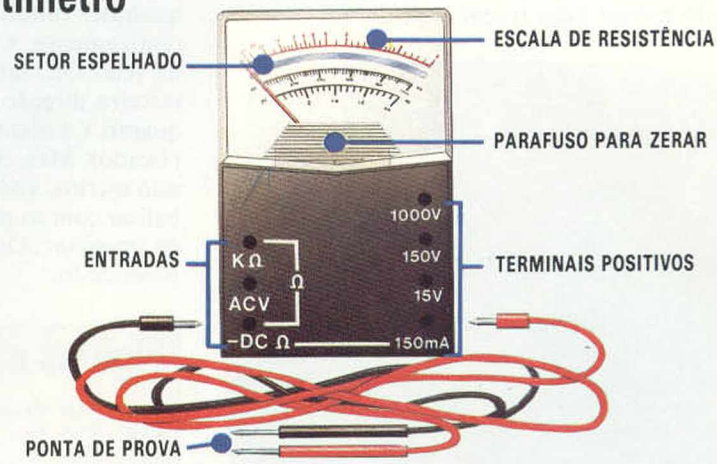
Você monta um aparelho semelhante com duas pilhas de 1,5 V, uma pequena lâmpada de 3 V e alguma extensão de fio. Alternativa melhor, no entanto, é um pequeno multímetro. Funcionando também como ohmímetro, ele pode testar não só a condutividade como a resistência. A unidade de medida da resistência é o ohm (símbolo Ω), nome do físico alemão George Ohm, que descobriu o fenômeno no século XIX.

A resistência é função da área da seção transversal do fio que conduz a corrente, mas também pode ser introduzida artificialmente, por meio de resistores.

Para nossa tarefa, uma ligação adequada oferece resistência mínima à passagem da pequena corrente que aciona o medidor ou aparelho de teste de condutividade. Assim, a luz deve brilhar com intensidade máxima ou a agulha do mostrador defletir totalmente. Qualquer reação em menor escala indica ligação malfeita.

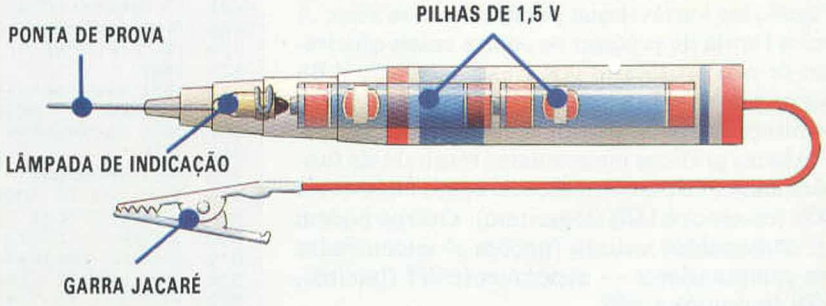
Além de medir condutividade, o multímetro tem duas funções: indica a corrente (em ampères) e a tensão (em volts). A diferença de potencial (tensão) entre dois pontos de um circuito que conduz uma corrente de 1 A e que dissipa 1 W de energia é de 1 V

Multímetro

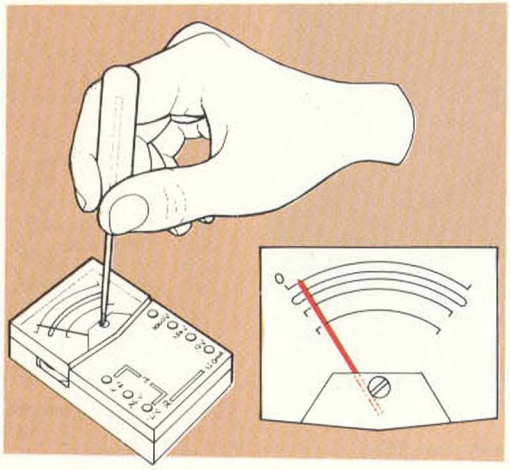


Variam muito os preços dos aparelhos que testam a condutividade e a resistência, medidas em ohms; a intensidade da corrente, medida em ampères; e a diferença de potencial, medida em volts. Mas só existem dois modos de apresentar seus resultados: o analógico e o digital. Instrumentos com bobina móvel que exibem seus dados por meio de uma agulha que se move (defletora) numa escala, de modo análogo ao aumento ou à diminuição do valor medido, são bem mais baratos que suas versões digitais.

Circuito de teste

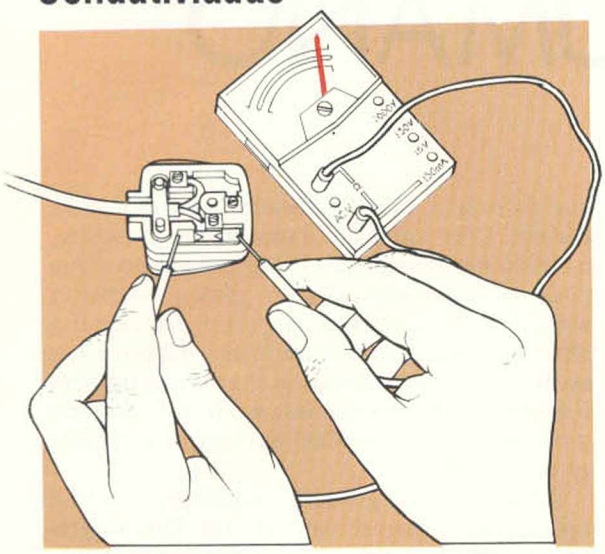


Zerando o medidor



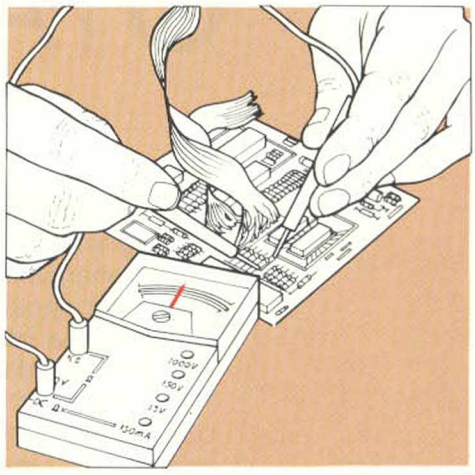
Variações e imprecisões
Medidores analógicos, nos quais a agulha de indicação se move pelo mostrador, devem permitir calibração. Em geral, isso se faz por meio de um parafuso colocado na base da agulha. O setor espelhado atrás da agulha dá ao observador a certeza de uma leitura acurada. Também o circuito que mede a resistência deve ser ajustável, para levar em conta outras variações e imprecisões.

Condutividade

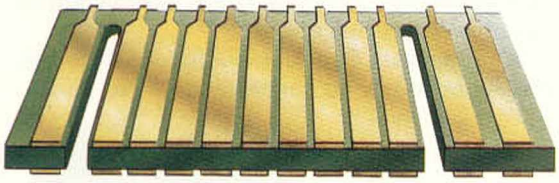


Condutividade e resistência
Um dos exemplos mais comuns de falta de condutividade ocorre nos fusíveis. O fusível é projetado para se queimar e abrir o circuito, quando se sobrecarrega. Na ausência de teste de condutividade, a solução usual é a troca do fusível velho por um novo.
O teste de condutividade indica também onde está o problema e o multímetro mede o valor da resistência.

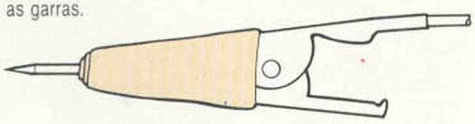
Resistência



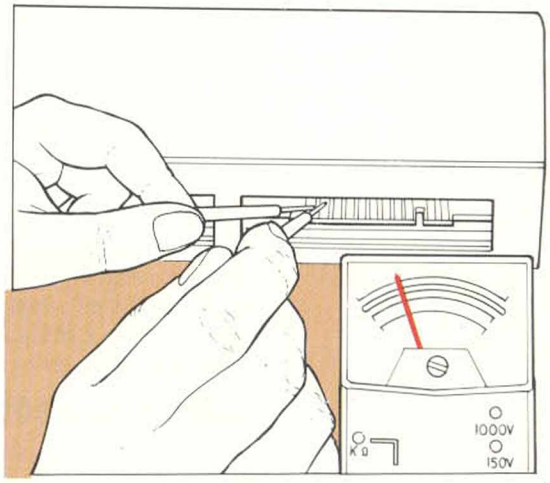
Conector de lâmina
A figura mostra o meio físico pelo qual os periféricos se conectam ao computador. Algumas máquinas, como o CP 200, têm somente essa entrada. Outras, como o CP 500, têm muitas. Conectores de lâmina são apenas um tipo de ligação de entrada e saída.



Garra jacaré
Esse instrumento impreciso converte-se numa boa ponta de prova quando se solda um pedaço pontiagudo de ferro entre as garras.

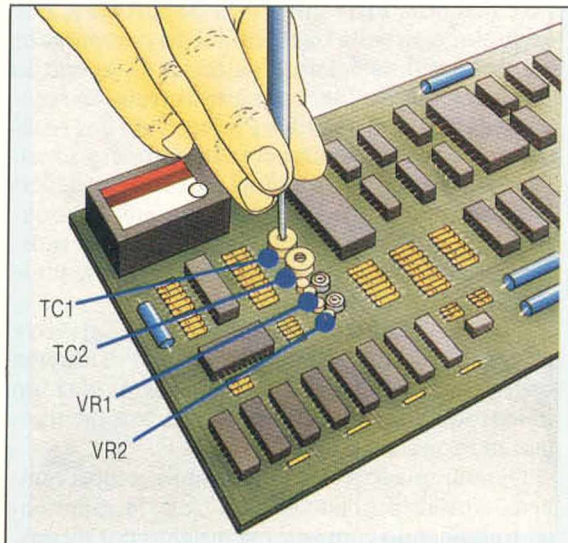


Voltagem
Um bom exemplo do uso do multímetro como instrumento de teste de voltagem: localizar a fonte de energia do conector de lâmina de seu computador. Consulte o manual e localize os pinos de 0, +5 e +12 V. Encoste o terminal negativo do medidor no pino de 0 V. A leitura resultante lhe dirá quão precisa é a regulagem do circuito de controle dentro do computador e seu suprimento de energia.



CUIDADO!

A garantia de seu microcomputador será invalidada se qualquer pessoa que não o fabricante ou seu agente autorizado abrir o gabinete.



Ajuste do Spectrum

Os proprietários dos primeiros Spectrum podem melhorar a qualidade da imagem da tevê. Dois resistores variáveis VR1 e VR2 controlam a proporção vermelho-verde e azul-amarelo, respectivamente.

Os capacitores TC1 e TC2 controlam o brilho dos caracteres e a intensidade das cores. Cinco parafusos mantêm unido o gabinete do Spectrum. São fáceis de localizar, na parte inferior do aparelho. Removendo-os, você expõe a PCI (Placa de Circuito Impresso). O procedimento para o ajuste dos aparelhos da linha Sinclair é bem semelhante, mas lembre-se do aviso ao lado, sobre a garantia de seu microcomputador.

O FATOR HUMANO

Um aspecto importante no projeto de programas é a interação homem/máquina. Os fatores que devem ser considerados ao se projetar essa interação são aqui investigados.

Durante muitos anos, a programação de computadores foi um tópico misterioso, entendido apenas por profissionais dedicados ao assunto. Antes do advento do microcomputador com teclado semelhante ao da máquina de escrever, dava-se entrada aos programas um byte por vez, acionando interruptores no painel frontal do computador ou perfurando fitas de papel.

O usuário de hoje é, ao contrário, uma criatura mimada. Os fabricantes não ficam esperando que ele esbarre na linguagem de máquina, tanto que a expressão "amigável ao usuário" *(user-friendly)* foi criada para indicar que os micros podem ser usados e programados por pessoas sem experiência. Em 1982, o Comitê Alvey, em seu relatório "Programa para tecnologia avançada da informação", identificou a ligação entre homem e máquina como uma das quatro principais áreas de pesquisa e desenvolvimento.

Influências sobre o diálogo

Além das restrições impostas pelos dispositivos de entrada e saída usados, o diálogo entre usuário e máquina sofre influência do software. Por exemplo, o sistema operacional do computador controla muitos detalhes da tela e da operação do teclado. A rapidez com que as teclas se repetem quando pressionadas e o intervalo das repetições são determinados pelo sistema operacional. Esse sistema também acumula toques, a fim de permitir ao computador estocar caracteres introduzidos mais rapidamente do que possam ser mostrados. Isso afeta a velocidade com que se pode entrar com informações.

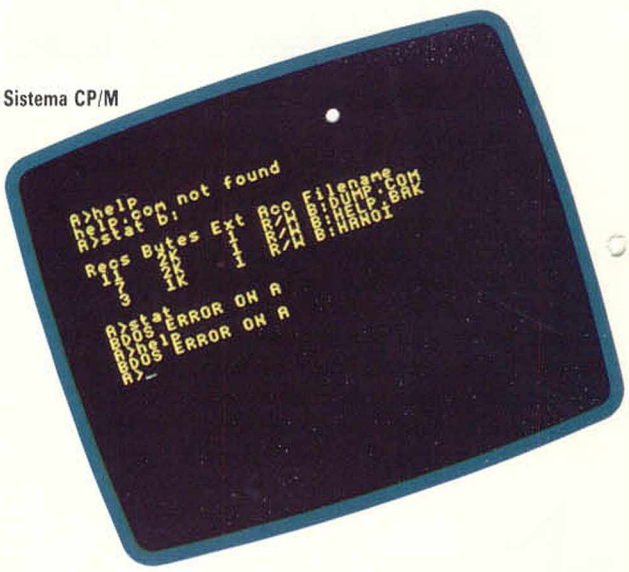
O tamanho do reservatório (buffer) é crítico e precisa ser conhecido pelo usuário — o sistema operacional CP/M, por exemplo, armazena um único toque; muitos microcomputadores acumulam dez toques ou mais.

Os acumuladores de toques, no entanto, também causam problemas. Um usuário experiente, trabalhando com sistema dirigido por menus, pode saber de antemão as escolhas de menu que fará. Por exemplo, 2, no menu principal, 5, no seguinte, e depois 3, 4 e 6. Graças a sua familiaridade com o sistema, ele digita essas escolhas com grande velocidade. Com um acumulador de dez caracteres, o usuário chega onde quer porque os toques são "lembrados" na seqüência correta. Com um acumulador de um caractere, o tempo necessário para expor o segundo menu pode ser mais longo do que o gasto para digitar a seqüência. Nesse caso, em vez de selecionar a alternativa número 5 desse menu, depois a 3, e assim por diante, apenas a alternativa número 6 é selecionada (porque esse é o único caractere guardado no acumulador) e o sistema pára nesse ponto.

Acumuladores grandes, contudo, também apresentam alguns inconvenientes. Um programa dirigido por menus que demore um tempo demasiado longo para reagir à digitação no teclado (isso ocorre se a escolha implica a leitura de um arquivo) pode fazer com que o usuário pense que nada está acontecendo. Sua reação natural será tentar novamente a última alternativa introduzida e pressionar várias teclas, até obter uma resposta. Muitas vezes, o programa tenta processar todos esses caracteres errados que ficam guardados no acumulador, oferecendo resultados surpreendentes.

A "limpeza", isto é, a eliminação dos registros da memória do computador a fim de liberar espaço para o trabalho é outra fonte de problemas. Tem-se a impressão de que o programa está parado por longo período, durante o qual podem ser tentadas ações corretivas. A limpeza, às vezes, causa problemas em programas grandes que lidam muito com strings (variáveis alfanuméricas).

Sistema CP/M



Algumas versões do BASIC permitem ao programador forçar uma limpeza; convém fazer isso a intervalos freqüentes e, nesse caso, o usuário deverá receber uma mensagem "por favor, es-

pere'' enquanto o computador estiver aparentemente inativo, mas procedendo à limpeza.

A maneira como a linguagem de programação lida com a entrada e a saída influencia o projeto da interface entre o computador e o usuário. Os bons recursos do BASIC para se lidar com strings permitem usá-las com mais sofisticação no diálogo entre homem e computador do que linguagens como PASCAL. Os BASICs dotados de comandos residentes para endereçamento do cursor proporcionam melhor diagramação *(layout)* de tela do que os outros. O mesmo se aplica para BASICs com comandos de gráficos.

O BASIC é bem suprido com comandos de entrada/saída — INPUT e PRINT são apropriados para programas simples. Mas, para controle efetivo da entrada (o tipo necessário para a produção de formulários que contenham campos protegidos para a entrada, por exemplo), experimente os GET$, INKEY$, INPUT$() e comandos similares. PRINT USING é um comando extremamente versátil para a formatação da saída; seu valor é inestimável para alinhar pontos decimais e ajustar colunas de texto.

Sistema Micronet 800

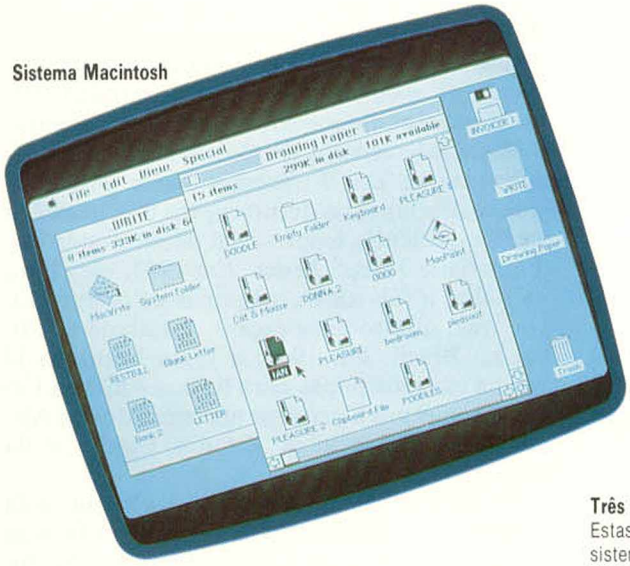
A psicologia do usuário

O usuário é o elemento mais imprevisível em todo sistema homem/máquina. Como qualquer outro componente, no entanto, ele possui certas características de desempenho que devem ser entendidas antes de se projetar a interface.

Usuário e computador são ''processadores de informação''. Contudo, os seres humanos têm limitações inerentes quanto à quantidade de informações novas que podem guardar na ''memória funcional'' — calcula-se que, para a maior parte dos tipos de informação, o cérebro pode guardar por volta de sete itens diferentes simultaneamente. O tamanho desses itens depende de quão significativos ou bem estruturados sejam. Se a informação consistir em caracteres aleatórios, cada item terá um caractere e nada mais. Mas, se, ao contrário, os caracteres formam palavras comuns, cada item lembrado pode ser uma palavra inteira. À medida que se amplia a estrutura das informações, aumenta a capacidade do usuário de lembrá-las e usá-las.

Alguns procedimentos ajudam a estruturar as informações. Um método consiste em relacionar os dados com estruturas familiares — assim funciona a metáfora da ''escrivaninha''; ou seja, vários tipos de informação — relatórios, gráficos etc. — são visualizados simultaneamente em janelas na tela, como os papéis na mesa de um executivo. Da mesma forma, um programa de planilha financeira pode ser organizado de modo a parecer um livro, com páginas e índices. Outro método é treinar o usuário para entender

Sistema Macintosh



estruturas não familiares. Por meio de exemplos repetidos e da explicação dos tópicos, o próprio programa ensina ao usuário como estruturar a informação. O inconveniente disso está no tempo e esforço despendidos. Instruções detalhadas e telas de ''ajuda'' fornecem um tipo de treinamento em linha, mas isso é difícil de usar com eficiência. Por fim, a apresentação da informação em padrões reconhecíveis constitui outro auxílio para o usuário, empregando-se cores ou a própria diagramação para ressaltar a informação desejada. Para ter idéia de como isso funciona, considere as instruções coloridas usadas em programas de videotexto. Numa página, ou tela, típica, o ''cabeçalho'' e o ''rodapé'' são blocos da mesma cor; há apenas uma cor de fundo e o texto é exibido em outras duas cores, que se alternam nos parágrafos. As palavras-chaves são enfatizadas pelo uso de outra cor. O propósito disso é permitir ao usuário selecionar somente a informação necessária. As instruções coloridas podem, no entanto, ser confusas se usadas em demasia, e alguns testes indicam que as pessoas desperdiçam muito tempo lendo e relendo parágrafos para tentar entender o significado das mudanças de cores. Uma boa regra é: nunca use mais de quatro cores de uma só vez.

Três graus
Estas fotos de telas de sistemas operacionais de microcomputadores ilustram três graus de interação com o usuário. Na primeira, um novato está tentando se comunicar com o sistema operacional CP/M. Mas este não tem recursos de ajuda ao usuário e, assim, exige conhecimentó prévio completo dos comandos. O segundo exemplo é de um sistema dirigido por menus — o menu inicial para o correio eletrônico Micronet 800 no BBC Micro. As opções são numeradas e o usuário faz sua opção entrando com o número apropriado. A tela não oferece muitas informações e, dessa forma, o usuário precisa entender as opções. A última foto mostra o sistema operacional do Macintosh, da Apple, que fornece indicações visuais e símbolos gráficos, assim como menus de simples e fácil compreensão.

SOLUÇÕES INTEGRADAS

Calculadoras portáteis e de mesa, máquinas de escrever e videogames, computadores — o nome Dismac se firma como uma solução nacional nos vários ramos da eletrônica.

A Dismac Industrial S.A. foi fundada em 1973, em Manaus, com o objetivo principal de conquistar uma fatia do mercado brasileiro de calculadoras eletrônicas, então dominado por equipamentos importados. Doze anos depois, 60% das calculadoras eletrônicas de mesa comercializadas no Brasil trazem a marca Dismac.

Esse desempenho decorreu, em boa parte, da vasta experiência empresarial de seu fundador, o austríaco Joseph Feder. Em 1947, ele criou, na Itália, a fábrica de máquinas de costura Elgin, que, quatro anos depois, instalaria sua filial no Brasil. Em 1969, a Elgin brasileira já vendia suas máquinas para toda a América Latina e começava a penetrar nos mercados da Alemanha, Grã-Bretanha, Estados Unidos e da própria Itália.

No final de 1972, Joseph Feder afastou-se da diretoria executiva da Elgin, passando a Dismac a mobilizar toda a sua atenção. Os resultados foram imediatos. Em 1973, surgiu a linha de calculadoras eletrônicas portáteis; em 1974, iniciou-se a produção de calculadoras eletrônicas de mesa, com doze dígitos de operação; em 1978, foram lançadas as caixas registradoras eletrônicas; em 1979, a Dismac tornou-se o maior exportador brasileiro de calculadoras eletrônicas e se preparava para novos empreendimentos, que culminaram, em 1981, com o D-8100, o primeiro micro fabricado no Brasil.

Tecnologia própria
Com suas linhas de produção instaladas em Manaus, a Dismac é uma das empresas da área de informática que apresentam os mais altos índices de nacionalização de seus produtos.

A versatilidade e o baixo preço em relação ao desempenho asseguraram rápida difusão ao D-8100. Trata-se de um microcomputador compatível com o Apple II, com 48 Kbytes de RAM e 12 Kbytes de ROM. Pode operar com televisor tanto em cores como em branco e preto ou com o monitor de vídeo; possui saída para áudio, interface para impressora e unidades de disco flexível de 5 1/4 polegadas. Em 1984, foram lançadas novas interfaces, permitindo a monitoração em vídeo colorido (placa PAL-M) e saída serial ou paralela para impressora.

Desenvolveu-se, a seguir, a linha de computadores. O Alfa 2000 e o Alfa 3000 são computadores profissionais de 64 Kbytes de RAM. O Alfa 2064 é um sistema multiusuário que permite a conexão de até oito terminais trabalhando em tempo real. A Dismac criou também uma série de pacotes aplicativos para a linha Alfa, possibilitando a utilização dos sistemas em qualquer escritório. Entre os aplicativos destacam-se os de folha de pagamento, faturamento, controle de estoques, contabilidade e crediário.

Em 1983, surgiram novos modelos de caixas registradoras eletrônicas, aptas a funcionar como terminais. Cada unidade pode operar em comunicação direta com o sistema central Dismac 3003 — um gerenciador da linha Alfa —, tornando-se um terminal automatizado de venda.

A busca de soluções integradas para facilitar o controle empresarial não impediu a Dismac de diversificar suas atividades. Ainda em 1983, foram lançados o Dismac Vídeo Game — que opera em cores no sistema PAL-M — e o Super Stick, um joystick de precisão, compatível com todos os aparelhos do sistema Atari.

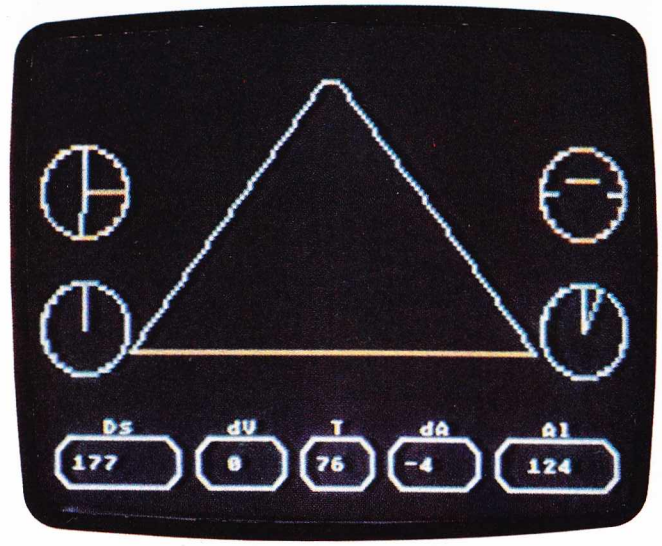
Em 1985, foi a vez da Olympia Dismac OAT 1200, a máquina de escrever eletrônica brasileira de maior capacidade de armazenamento de dados do mercado: armazena até 1.000 caracteres em sua memória. Além disso, é a única a possuir memória partilhada (em até dez compartimentos), o que torna possível gravar, em separado, cabeçalhos, endereços e frases inteiras habitualmente utilizadas.

A Dismac respondia, em 1985, por mais de 30% do mercado nacional de caixas registradoras eletrônicas e por quase 20% do mercado de computadores. Na segunda metade dos anos 80, ela pretendia repetir esse desempenho favorável nos campos do lazer eletrônico e das máquinas de escrever ultra-sofisticadas.

bater na sinalização da pista. O delta de velocidade (dV) assinala a taxa de variação da velocidade por segundo (5 a −5), e o de altitude (dA) faz o mesmo em relação à ascensão e à descida, variando de −25 a 25 pés por segundo (a razão de descida não pode exceder −4 pés/s). Por fim, um cronômetro marca o tempo de aterrissagem. Para conseguir uma aterrissagem curta antes do final da pista, mantenha o dV ao mínimo (−5).

Controle de movimentos: ascensão, puxe o joystick; descida, empurre-o; aumento da velocidade, vire-o para a esquerda; redução, vire-o para a direita. Os pontos são calculados considerando-se o tempo de aterrissagem, o planeio e a distância percorrida na pista.

```
1  REM     * * * * * * * * * * * *
2  REM     * PARA LINHA  APPLE  *
3  REM     *                    *
4  REM     *     VOO NOTURNO    *
5  REM     * * * * * * * * * * * *
6  REM
9  HOME : HGR2 : POKE 49233,0: POKE
     49236,0: FOR I = 1 TO 100
10  HCOLOR=  INT ( RND (1) * 8): HPLOT
     140,80 TO  INT ( RND (1) * 2
     80), INT ( RND (1) * 160): NEXT
     I
11  POKE 799,0: POKE 800,0: FOR I
      = 770 TO 795: READ M: POKE
     I,M: NEXT I
12  DATA 172,1,3,174,1,3,169,4,32
     ,168,252,173,48,192,232,208,
     253,136,208,239,206,0,3,208,
     231,96
13  CLEAR
14  HOME
15 HS = 0: DIM B$(35)
16  HGR
17 DEG = .017452406: GOSUB 800
50 SH = 160:SW = 280:RL = 4000:RW
     = 60
52 T = 2
54 FS = 66 / 45
90 LD = 0:KI = 80:VZ = 0.85:DZ =
     100:AZ = 5:XV = 30:YV = 80:G
     X = 0:XD = 30:YD = 120:XA =
     250:YA = 120
100 D = 15000 +  RND (1) * 15000
102 A = 900 +  RND (1) * 900
103  IF ST = 0 THEN D = 20000:A =
     1200
105 TX = D / 150
130 RC = 0:V = 200:PR = 0:CR = 0
140 BD = 0:BA = 0
150 RL = 4000:RW = 60
200  REM
208  IF PR = 0 AND D < 4000 AND A
      > 1000 THEN B$ = "MUITO ALT
     O P/ UMA ATERRAGEM SEGURA": GOTO
     500
210  IF A > 1 AND V < 80 THEN B$ =
     "VELOCIDADE ABAIXO DE STALL
     (80)": GOTO 500
211  IF A < 30 AND PR = 0 THEN B$
      = "ALTITUDE ABAIXO DO MINIM
     O (30)": GOTO 500
212  IF D < 1 AND PR = 0 THEN D =
     4000:PR = 1
213  IF D < 1 THEN B$ = "BATEU NO
      FIM DA PISTA.": GOTO 500
214  IF PR = 1 THEN 300
215 X =  INT (D / 1000): IF X < 4
      THEN 218
216  IF X >  = KI THEN 400
218 KI = X: GOSUB 920
220 A1 = T * A / D * SH:A2 = ((T *
     A) / (D + RL)) * SH
230 RF = T * RW / D * SW:L1 = (SW
     - RF) / 2:R1 = L1 + RF
240 RR = ((T * RW) / (D + RL)) *
     SW:L2 = (SW - RR) / 2:R2 = L
     2 + RR
248  IF D < 4000 THEN  GOSUB 900:
      GOTO 400
250  GOSUB 900
260  GOTO 400
300  GOSUB 920
310 A1 = 160:A2 = ((T * A) / D) *
     SH: IF A2 < 1 THEN A2 = 1
315  IF A < RF THEN RF = RF + 1: IF
     RF > 278 THEN RF = 278
316  IF A > RF THEN RF = RF - 1: IF
     RF < RR THEN RF = RR + 7
320 L1 = (SW - RF) / 2:R1 = L1 +
     RF
325 RR = ((T * RW) / (D)) * SW:L2
      = (SW - RR) / 2:R2 = L2 + R
     R
330  GOSUB 900
350  IF A > D / 10 THEN B$ = "MUI
     TO ALTO P/ UMA ATERRAGEM SEG
     URA": GOTO 500
400 T1 = TI:TI = TI + 1:TD = TI -
     T1
403 X =  PDL (0):Y =  PDL (1): IF
     Y > 50 AND Y < 200 AND X > 5
     0 AND X < 200 THEN 450
405  IF X > 200 AND Y > 200 THEN
     VD = VD + 1:RC = RC - V / 40
     : GOTO 440
406  IF X > 200 AND Y < 50 THEN V
     D = VD + 1:RC = RC + V / 40:
      GOTO 440
407  IF X > 200 AND Y > 50 AND Y <
     200 THEN VD = VD + 1: GOTO 4
     40
409  IF X < 50 AND Y > 200 THEN V
     D = VD - 1:RC = RC - V / 40:
      GOTO 440
410  IF X < 50 AND Y < 50 THEN VG
     D = VD - 1:RC = RC + V / 40:
      GOTO 440
411  IF X < 50 AND Y > 50 AND Y <
     200 THEN VD = VD - 1: GOTO 4
     40
413  IF X > 50 AND X < 200 AND Y >
     200 THEN RC = RC - V / 40: GOTO
     440
414  IF X > 50 AND X < 200 AND Y <
     50 THEN RC = RC + V / 40: GOTO
     440
440  IF VD <  - 5 THEN VD =  - 5
442  IF VD > 5 THEN VD = 5
444  IF RC > 25 THEN RC = 25
```

Simulador emocionante
Este programa simula uma situação de emergência, com dois níveis de emoção. Você pode escolher uma situação conhecida, treinar bastante e, daí, enfrentar o modo mais difícil, com distância da pista e altitude aleatórias. Esse é o melhor caminho para um bom desempenho, que, segundo os especialistas, gira em torno de 2.500 pontos.

```
446  IF RC <  - 25 THEN RC =  - 2
     5
450  IF A = 0 THEN 452
451 A = A + RC: IF A > 1800 THEN
     A = 1800
452 AD = (D) * 0.06: IF PR = 1 THEN
     460
453  IF  ABS (A - AD) < 30 THEN G
     P = 0: GOTO 460
454 GP =  - (A - AD) / 30
455  IF GP > 12 THEN GP = 12
456  IF GP <  - 12 THEN GP =  - 1
     2
459 CR = CR - 5
460 DV = FS * V * TD:D = D - DV
470 V = V + VD: IF V > 300 THEN V
     = 300
471  IF A < 0 THEN A = 0
472  IF A > 0 THEN 480
473  IF RC <  - 4 THEN B$ = "EXCE
     SSIVA ELEVACAO (-4) BATEU.":
      GOTO 500
475 V = V + VD - 3:RC = 0: IF V <
     1 THEN 600
476  IF LD = 0 THEN LD = 1: FOR I
      = 1 TO 10: POKE 768,1: POKE
     769,1: CALL 770: NEXT
480  HCOLOR= 0: HPLOT 30,120 TO X
     D,YD
481  HCOLOR= 3:I = D / DZ - 90: GOSUB
     999: HPLOT 30,120 TO X + 30,
     Y + 120:XD = X + 30:YD = Y +
     120
482  HCOLOR= 0: HPLOT 250,120 TO
     XA,YA
483  HCOLOR= 3:I = A / AZ - 90: GOSUB
     999: HPLOT 250,120 TO X + 25
     0,Y + 120:XA = X + 250:YA =
     Y + 120
484  HCOLOR= 0: HPLOT 30,80 TO XV
     ,YV
485  HCOLOR= 3:I = V / VZ - 90: GOSUB
     999: HPLOT 30,80 TO X + 30,Y
      + 80:XV = X + 30:YV = Y + 8
     0
487  HCOLOR= 0: HPLOT 241,80 + GX
      TO 259,80 + GX
488  HCOLOR= 3: HPLOT 241,GP + 80
      TO 259,GP + 80:GX = GP
490  VTAB 23: HTAB 5: PRINT  INT
     (D);" ";
491  HTAB 31: IF A < 100 THEN  INVERSE
     : PRINT "*";: NORMAL : PRINT
     " ";
492  PRINT  INT (A);" ";
493  HTAB 15: PRINT  INT (VD);" "
     ;
494  HTAB 24: PRINT  INT (RC);" "
     ;
498  HTAB 19: PRINT  INT (TI);
499  GOTO 200
500  REM
505  VTAB 21: HTAB 3: PRINT "
                "
510  VTAB 21: HTAB 4: PRINT B$;
515  FOR I = 1 TO 10: POKE 768,1:
      POKE 769,10: CALL 770: NEXT

520  FOR I = 1 TO 50
525  POKE 49234,0: POKE 49237,0: POKE
     49236,0: POKE 49235,0
526  NEXT I
550  GOTO 650
602 PT = (TX - TI) * 10 + D / 2 +
     CR + 1000
606  IF PT > HS THEN HS =  INT (P
     T): POKE 800, INT (HS / 256)
     : POKE 799,HS -  INT (HS / 2
     56) * 256
610  VTAB 21: HTAB 4
615  PRINT "PONTOS "; INT (PT);"
     "
650  VTAB 24: HTAB 11: PRINT "PRE
     SSIONE BUTTON  0";
655  POKE 769, RND (1) * 254 + 1:
      POKE 768,3: CALL 770
660  IF  PEEK ( - 16287) > 127 THEN
     4
```

```
661  FOR I = 1 TO 250: NEXT I: IF
      PEEK ( - 16287) > 127 THEN
     4
662  POKE 769, RND (1) * 254 + 1:
      POKE 768,3: CALL 770
664  HTAB 11: INVERSE : PRINT "PR
     ESSIONE BUTTON  0":: NORMAL
     : FOR I = 1 TO 250: NEXT I
666  GOTO 650
800  VTAB 22: HTAB 4
802  PRINT "/------"; CHR$ (92);"
     /----"; CHR$ (92);"/---"; CHR$
     (92);"/----"; CHR$ (92);"/--
     ----"; CHR$ (92)
803  HTAB 4: PRINT "!       !!
     !!    !!     !!       !"
804  HTAB 4: PRINT  CHR$ (92);"--
     ----/"; CHR$ (92);"----/"; CHR$
     (92);"---/"; CHR$ (92);"----
     /"; CHR$ (92);"------/";
806 HS =  PEEK (800) * 256 +  PEEK
     (799)
818  VTAB 21: HTAB 4: PRINT "VOO
     NOTURNO     PONTUACAO MAX. ";
     HS;
820 R = 19
822  FOR I = 0 TO 360 STEP 3
824 X = R *  COS (I * DEG):Y = R *
      SIN (I * DEG) * .9
825  HCOLOR= 3: HPLOT X + 30,Y +
     120: HPLOT X + 250,Y + 120
826  HCOLOR= 3: HPLOT X + 30,Y +
     80: HPLOT X + 250,Y + 80
828  NEXT I
832  HCOLOR= 3: HPLOT 30,80 TO 47
     ,80
833  HCOLOR= 3: HPLOT 30,80 TO 30
     ,63
834  HPLOT 232,80 TO 240,80: HPLOT
     260,80 TO 268,80
836  HPLOT 30,120 TO 30,102: HPLOT
     250,120 TO 250,102
860  VTAB 21: HTAB 3
862  PRINT "COMECO NORMAL(C) OU A
     LEATORIO (B) ?"
870  IF  PDL (1) < 50 THEN ST = 1
     : GOTO 880
872  IF  PDL (1) > 210 THEN ST =
     0: GOTO 880
875  POKE 769, RND (1) * 254 + 1:
      POKE 768,3: CALL 770
876  GOTO 870
880  VTAB 21: HTAB 3: PRINT "
                ";
881  VTAB 21: HTAB 4
882  PRINT "   DS      DV     T
     DA       AL";
899  RETURN
900  REM
901  IF L1 < 50 THEN L1 = 50
902  IF L2 < 50 THEN L2 = 50
903  IF L2 > 230 THEN L2 = 230
904  IF R2 > 230 THEN R2 = 230
905  IF R1 < 50 THEN R1 = 50
906  IF R1 > 230 THEN R1 = 230
907  IF A1 > 160 THEN A1 = 160
908  IF A2 > 160 THEN A2 = 160
910  HCOLOR= 3: IF PR = 1 THEN 91
     4
912  HPLOT L1,A1 TO R1,A1
914  HPLOT L2,A2 TO R2,A2
915  HCOLOR= 3: HPLOT L1,A1 TO L2
     ,A2: HPLOT R1,A1 TO R2,A2
918  RETURN
920  HCOLOR= 0: HPLOT L1,A1 TO R1
     ,A1: HPLOT L2,A2 TO R2,A2
922  HPLOT L1,A1 TO L2,A2: HPLOT
     R1,A1 TO R2,A2: RETURN
999 X = 17 *  COS (I * DEG):Y = 1
     7 *  SIN (I * DEG) * .9: RETURN
```


VÔO NOTURNO
PARA LINHA APPLE

Você está nos controles de um avião em plena noite e tudo o que pode ver são as luzes do painel de instrumentos e a fraca iluminação da pista de pouso à distância. Sua missão é aterrissar com segurança. Pode escolher a forma de começar: pressionando o joystick para baixo, o início será normal — a 20.000 pés de distância da pista e a uma altitude de 1.200 pés —; movendo-o para cima, a distância da pista e a altitude serão aleatórias, exigindo rápida e precisa interpretação dos instrumentos. Este programa é um simulador de vôo. O susto fica por conta de sua imaginação. Boa aterrissagem.

O mostrador superior esquerdo da tela indica sua velocidade. Mantenha-a acima de 80 km/h enquanto não tocar o solo; do contrário, o avião entrará em stall e cairá. Embaixo, à esquerda, encontram-se os mostradores numérico (DS) e de distância da pista (após a aterrissagem passam a indicar a distância até o fim da pista). O indicador de planeio, no canto superior direito, mostra um sinal laranja, que deve ser mantido entre os dois marcadores. Embaixo, situa-se o altímetro e seu mostrador numérico (AL). Procure ficar no mínimo a 30 pés para não